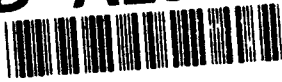


AD-A239 436



**DEFENSE SYSTEMS
SUPPORT ORGANIZATION**

FUNCTIONAL DESCRIPTION
FD 4-91
14 JUNE 1991

**FUNCTIONAL DESCRIPTION
FOR THE
MAPPING AND GRAPHIC INFORMATION CAPABILITY
(MAGIC)**

DTIC
ELECTRONIC
AUG 14 1991
S B D

APPROVED FOR
PUBLIC RELEASE
DISTRIBUTION UNLIMITED

91-07571



91 8 12 060

91 8 12 035

RECORD OF CHANGES

Change Number	Dated	Date Entered	Signature of Person Making Change

DCA FORM 65
MAR 87

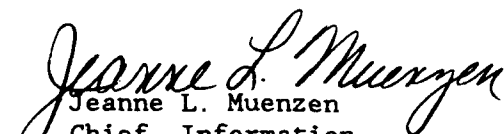
DEFENSE SYSTEMS SUPPORT ORGANIZATION

Functional Description FD 4-91

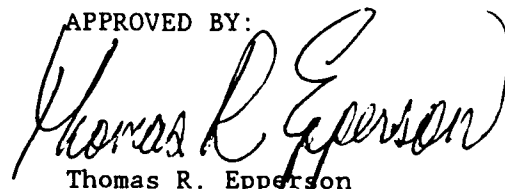
14 June 1991

FUNCTIONAL DESCRIPTION
OF THE
MAPPING AND GRAPHIC INFORMATION CAPABILITY (MAGIC)

SUBMITTED BY:


Jeanne L. Muenzen
Chief, Information
Systems Branch

APPROVED BY:


Thomas R. Epperson
Deputy Director
NMCS ADP Directorate

Copies of this document may be obtained from the Defense Technical Information Center, Cameron Station, Alexandria, Virginia 22304-6145.

Approved for public release; distribution unlimited.

ACKNOWLEDGMENT

This Functional Description (FD) was prepared under the general direction of the Chief, Information Systems Branch (JNGG); the Chief, General Applications Division (JNG); and the Deputy Director, National Military Command System (NMCS) Automated Data Processing (ADP) Directorate (JN).



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail. and/or Special
A-1	

CONTENTS

Section		Page
	ACKNOWLEDGMENT	ii
	ABSTRACT	viii
1.	GENERAL	1-1
1.1	Purpose of the Functional Description	1-1
1.2	Project References	1-1
1.3	Terms and Abbreviations	1-3
2.	SYSTEM SUMMARY	2-1
2.1	Background	2-1
2.1.1	Current System	2-1
2.1.1.1	Graphic Information Presentation System (GIPSY)	2-1
2.1.1.2	GIPSYmate	2-2
2.1.2	Proposed System	2-3
2.2	Objectives	2-3
2.3	Existing Methods and Procedures	2-4
2.3.1	Capabilities of the System	2-5
2.3.2	Structure of the System	2-5
2.3.2.1	System Initiation	2-5
2.3.2.2	Syntax Processing	2-5
2.3.2.3	Data Selection	2-9
2.3.2.4	Matrix Generation	2-9
2.3.2.5	Statistical Reports Processing	2-9
2.3.2.6	Geographic Processing	2-9
2.3.2.7	Formatted Reports Processing	2-14
2.3.2.8	User Calls Processing	2-14
2.3.2.9	Batch Processing	2-14
2.3.2.10	Related Structures	2-14
2.3.3	System Specification	2-20
2.3.4	Deficiencies and Limitations	2-22
2.4	Proposed Methods and Procedures	2-24
2.4.1	Summary of Improvements	2-24
2.4.2	Summary of Impacts	2-25
2.4.2.1	User Organizational Impacts	2-25
2.4.2.2	User Operational Impacts	2-25
2.4.2.3	User Development Impacts	2-26
2.5	Assumptions and Constraints	2-26
2.5.1	Assumptions	2-26
2.5.1.1	User Assumptions	2-26
2.5.1.2	System Assumptions	2-26
2.5.1.3	System Maintenance Assumptions	2-27
2.5.1.4	System Developmental Assumptions	2-27
2.5.1.5	Hardware Assumptions	2-28
2.5.2	Constraints	2-28

Section		Page
3.	DETAILED CHARACTERISTICS	3-1
3.1	Specific Performance Requirements	3-1
3.1.1	Accuracy and Validity	3-1
3.1.1.1	System Accuracy	3-1
3.1.1.1.1	Correct User Command Processing	3-1
3.1.1.1.2	Data Type Checking	3-1
3.1.1.1.3	Map Data Accuracy	3-1
3.1.1.1.4	Computational Accuracy	3-1
3.1.1.2	System Validity	3-2
3.1.1.2.1	Valid User Command Processing	3-2
3.1.1.2.2	User Data Considerations	3-2
3.1.2	Timing	3-2
3.1.3	Capacity Limits	3-2
3.2	Functional Area System Functions	3-2
3.2.1	Graphical User Interface	3-2
3.2.1.1	Context-Sensitive Help	3-3
3.2.1.2	Error Detection/Handling	3-3
3.2.2	Data Retrieval	3-4
3.2.3	Business Reports	3-4
3.2.4	Geographic Displays	3-5
3.2.5	Graphic Editing	3-6
3.2.6	Slide Presentations	3-6
3.2.7	Internal System Functions	3-6
3.2.8	System Programmer Functions	3-7
3.3	Inputs and Outputs	3-8
3.4	Database/Data Bank Characteristics	3-8
3.4.1	Host-Based User Data	3-8
3.4.2	Workstation-Based Databases	3-8
3.4.2.1	Oracle Relational Database Management System (RDBMS)	3-8
3.4.2.2	DeLorme Mapping System (DMS)	3-9
3.5	Failure Contingencies	3-9
4.	DESIGN CONSIDERATIONS	4-1
4.1	System Description	4-1
4.2	System Functions	4-1
4.2.1	Performance Requirements	4-1
4.2.1.1	Accuracy and Validity	4-2
4.2.1.2	Timing	4-2
4.2.1.3	Capacity Limits	4-2
4.2.2	Functional Requirements	4-2
4.2.2.1	Human Interface	4-2
4.2.2.2	Data Management	4-3
4.2.2.3	Business Graphics	4-3
4.2.2.4	Geographic Mapping	4-3
4.2.2.5	Graphic Editor	4-4
4.2.2.6	Slide Show	4-4
4.2.2.7	Internal Processing	4-4
4.2.2.8	Programmer Utilities	4-4

Section		Page
4.3	Flexibility	4-4
4.4	System Data	4-5
5.	ENVIRONMENT	5-1
5.1	Equipment Environment	5-1
5.1.1	Current Hardware Environment	5-1
5.1.1.1	Supported Terminal Types	5-1
5.1.1.2	On-Line Printer	5-2
5.1.2	Proposed Hardware Environment	5-2
5.1.2.1	Prototype Environment	5-2
5.1.2.2	Target Environment	5-3
5.2	Support Software Environment	5-3
5.2.1	Prototype Environment	5-4
5.2.2	Target Environment	5-5
5.3	Communications Requirements	5-6
5.3.1	Graphic Overview	5-6
5.3.2	Hardware	5-6
5.3.3	Software	5-9
5.4	Interfaces	5-9
5.4.1	Oracle RDBMS	5-10
5.4.2	Wingz	5-10
5.4.3	DeLorme Mapping Systems	5-10
5.4.4	Graphic Information Presentation System (GIPSY)	5-10
5.5	Summary of Impacts	5-11
5.5.1	ADP Organizational Impacts	5-11
5.5.2	ADP Operational Impacts	5-11
5.5.3	ADP Development Impacts	5-11
5.6	Failure Contingencies	5-11
5.7	Assumptions and Constraints	5-12
6.	SECURITY	6-1
6.1	Background Information	6-1
6.2	Control Points, Vulnerabilities, and Safeguards	6-1
6.2.1	Control Points	6-1
6.2.1.1	Input Control Points	6-1
6.2.1.2	Process Control Points	6-2
6.2.1.3	Output Control Points	6-2
6.2.2	Vulnerabilities	6-2
6.2.3	Safeguards	6-3
6.2.3.1	Administrative Safeguards	6-3
6.2.3.2	Physical Safeguards	6-3
6.2.3.3	Technical Safeguards	6-3
6.3	System Monitoring and Auditing	6-4
6.3.1	Journalizing	6-4
6.3.1.1	Triggering Criteria	6-4
6.3.1.2	Identification Information	6-4
6.3.1.3	Application Data	6-5
6.3.1.4	Journal Use	6-5

Section		Page
6.3.2	Audit Trail	6-5
7.	SYSTEM DEVELOPMENT PLAN	7-1
8.	COST FACTORS	8-1
8.1	The REVIC Model	8-1
8.1.1	REVIC vs. COCOMO	8-1
8.1.2	Calibration and Accuracy	8-2
8.2	Sizing Estimation	8-2
8.3	Ada Language Alternative	8-5
8.3.1	Development Costs	8-5
8.3.2	Maintenance Costs	8-5
8.3.3	Risk Analysis	8-5
8.4	C Language Alternative	8-10
8.4.1	Development Costs	8-10
8.4.2	Maintenance Costs	8-10
8.4.3	Risk Analysis	8-10
8.5	Conclusions	8-10

APPENDIXES

A.	Functional System Requirements	A-1
	DISTRIBUTION	B-1
	STANDARD FORM 298	C-1

ILLUSTRATIONS

Figure		Page
2-1	GIPSY's System Organization	2-6
2-2	Executive Module	2-7
2-3	Syntax Module	2-8
2-4	Data Selection Module	2-10
2-5	Matrix Generation Module	2-11
2-6	Statistical Reports Module	2-12
2-7	Geographic Reports Module	2-13
2-8	Generalized Data Reports Module	2-15
2-9	Call Module	2-16
2-10	Batch GIPSY Module	2-17
2-11	GIPSY's System Configuration at the DSSO	2-21
5-1	Prototype Communications Requirements	5-7
5-2	Target Communications Requirements	5-8

TABLES

Number		Page
8-1	C Language Sizing Estimation	8-3
8-2	Ada Language Sizing Estimation	8-4
8-3	Environmental Factors for Ada Development	8-6
8-4	Estimated Development Costs Using Ada	8-7
8-5	Estimated Maintenance Costs Using Ada	8-8
8-6	Ada Risk Analysis	8-9
8-7	Environmental Factors for C Development	8-11
8-8	Estimated Development Costs Using C	8-12
8-9	Estimated Maintenance Costs Using C	8-13
8-10	C Risk Analysis	8-14

ABSTRACT

This Functional Description (FD) identifies preliminary design considerations, performance requirements, and user impacts. It also defines the requirements to be satisfied and provides the users with a clear statement of the operational capability to be developed. The FD is the basis for mutual understanding between the user and the developer. It outlines the plan for development and implementation of the Mapping and Graphic Information Capability (MAGIC) and provides a summary of cost factors.

The FD is divided into six major sections. These sections cover System Summary (Section 2), Detailed Characteristics (Section 3), Design Considerations (Section 4), Environment (Section 5), Security (Section 6), and Cost Factors (Section 8).

This document supersedes the Functional Description for the Graphic Information Presentation System (GIPSY) (configuration identifier CBSI-88-001-MFD/8719) that was delivered under DCA Contract DCA100-87-C-0064 and dated 1 February 1988.

SECTION 1. GENERAL

This section provides a general introduction to the document by describing its purpose, listing project references, and including a terms and abbreviations subsection.

1.1 Purpose of the Functional Description

This Functional Description (FD) for the Mapping and Graphic Information Capability (MAGIC) is written to provide:

- a. A bridging document that clearly delineates and documents both the structure and functionality of the current Graphic Information Presentation System (GIPSY) while providing complete and yet, concise information linking those functions to the architecture and components of MAGIC
- b. Documentation of all requirements (system, functional, user, performance, etc.) that MAGIC will satisfy such that, this FD shall serve as a basis for mutual understanding between the Government and the contractor
- c. A basis for discussion of planned system processing, assumptions, constraints, and expected impacts upon users
- d. Preliminary, top-level design information related to MAGIC
- e. A basis for the development of system integration tests and formal qualification testing (FQT).

1.2 Project References

The following references were used in the preparation of this document:

- a. Air Force Contract Management Division (AFCMD), REVIC Software Cost Estimating Model User's Manual, Version 8.7.6, Kirtland AFB, NM, August 1989
- b. American National Standards Institute (ANSI), Programming Language C, ANSI Standard X3.159-1989, New York, NY, 14 December 1989
- c. ANSI, Reference Manual for the Ada Programming Language, ANSI Standard ANSI/MIL-STD-1815A-1983, New York, NY, 17 February 1983
- d. Department of Defense (DoD), Defense System Software Development, Department of Defense Standard DoD-STD-2167A, Washington, D.C., 29 February 1988

- e. DoD, DoD Automated Information Systems (AIS) Documentation Standards, Department of Defense Standard DoD-STD-7935A, Washington, D.C., 31 October 1988
- f. DoD, Economic Analysis and Program Evaluation for Resource Management, Department of Defense Instruction (DoDI) 7041.3, Washington, D.C., 18 October 1972
- g. DoD, Functional Description, Data Item Description (DID) DI-IPSC-80689, Washington, D.C., 31 October 1988
- h. Joint Data Systems Support Center (JDSSC), Documentation Standards and Publications Style Manual, Procedures Manual (PM) 1-90, Washington, D.C., 1 August 1990
- i. JDSSC, GIPSYmate Software User's Manual, Software User's Manual (SUM) 1-91, Washington, D.C., 28 January 1991
- j. JDSSC, Graphic Information Presentation System (GIPSY) Programmers Reference Manual, Washington, D.C., 31 October 1986 (Draft)
- k. JDSSC, Graphic Information Presentation System (GIPSY) Test Procedures, Technical Memorandum (TM) 303-91, Washington, D.C., 1 February 1991
- l. JDSSC, Users Manual for the Graphic Information Presentation System (GIPSY), Users Manual (UM) 7-91, Washington, D.C., 1 February 1991
- m. JDSSC, Operational Concept Document for the Mapping and Graphic Information Capability (MAGIC), Technical Memorandum (TM) 406-91, Washington, D.C., 15 January 1991
- n. JDSSC, Software Development Plan (SDP) for the Mapping and Graphic Information Capability (MAGIC), SDP 2-90, Washington, D.C., 1 November 1990
- o. JDSSC, Software Quality Program Plan for the Mapping and Graphic Information Capability (MAGIC), Washington, D.C., 23 July 1990 (Draft)
- p. JDSSC, Software Standards and Procedures Manual for the JNGG Graphics Program, Technical Memorandum (TM) 405-90, Washington, D.C., 1 December 1990
- q. JDSSC, System Design Document for the Modernized GIPSYmate System, System Planning Manual (SPM) System Specification (SS) 161-88, Washington, D.C., 14 July 1988

- r. National Bureau of Standards (NBS), POSIX: Portable Operating System Interface for Computer Environments, Federal Information Processing Standards Publication (FIPS PUB) 151, Washington, D.C.,
12 September 1988.

1.3 Terms and Abbreviations

The following terms, abbreviations, and acronyms specific to this FD are listed below:

ADP	-----	Automated Data Processing
ANMCC	-----	Alternate National Military Command Center
ANSI	-----	American National Standards Institute
ASCII	-----	American Standard Code for Information Interchange
BGIPSY	-----	The Batch GIPSY Module
BIS-7705	-----	The GIPSYmate Interface between the WWS and GIPSY
BOOKFILE	-----	Holds the BOOK (formatted report) created by the GDRMOD module
CalComp	-----	CalComp, Inc. of Anaheim, California; developers and distributors of color electrostatic plotters and their software
CALMOD	-----	The Call Module in GIPSY
CBSI	-----	Computer Based Systems, Incorporated
CCTC	-----	Command and Control Technical Center (forerunner of JDSSC)
CDRL	-----	Contract Data Requirements List
CMDLIB	-----	The Time Sharing Command Library
COTS	-----	Commercial Off-The-Shelf Software
CSCI	-----	Computer Software Configuration Item
CUELIB	-----	The GIPSY Cue Library of help and error messages
DAFC	-----	Directive Action File Control
DATSEL	-----	The GIPSY Data Selection Module
DC	-----	Device Coordinate
DCA	-----	Defense Communications Agency
DID	-----	Data Item Description
DIF	-----	Data Interchange Format
DISPLA	-----	The GIPSY Statistical Graphics Module
DMS	-----	DeLorme Mapping System
DoD	-----	Department of Defense
DoDI	-----	Department of Defense Instruction
DPS8000	-----	Distributed Processing System #8000
DRL	-----	Time Sharing System Derail
EDIT	-----	Honeywell H6000/DPS8000 Time Sharing System Editor
ENVIRO	-----	The ..ENVIRO file used to hold terminal IDs and their associated GIPSY device code
EOM	-----	End-of-Memory
ETC	-----	Enhanced Terminal Capability; the optional GIPSYmate Interface Package developed by the Government
EXEC	-----	The GIPSY Executive Module
FD	-----	Functional Description
FDT	-----	File Descriptor Table

Font ----- A character set of the same size and face
 FORT ----- TSS Fortran Interactive Compiler
 FORTRAN 66 ----- The 1966 Honeywell H6000/DPS8000 version of FORTRAN
 FQT ----- Formal Qualification Testing
 GCFILE ----- The GIPSY Command File
 GCOS3 ----- General Comprehensive Operating Supervisor
 GCOS8 ----- Virtual General Comprehensive Operating Supervisor
 GDRMOD ----- The GIPSY Generalized Data Reports Module
 GDS ----- Graphic Data Set
 GEOMOD ----- The GIPSY Geographic Mapping Module
 GIPSY ----- Graphic Information Presentation System
 GIPSYmate ----- The PC-based system used to provide access to the host-based
 GIPSY on the WIS/CUC and the Z-248 PC/AT
 GKS ----- Graphical Kernel System
 GMAP ----- General Macro Assembler Program; Honeywell's assembler
 language
 GYSTAT ----- The GIPSY Statistics File
 HIS ----- Honeywell Information Systems
 HOL ----- High Order Language (e.g., Ada, FORTRAN, Pascal)
 H6000 ----- Honeywell 6000 Mainframe computer standard at WWMCCS sites;
 now a DPS8000
 H* ----- A Honeywell program link file; GIPSY currently uses H* files
 to store all GIPSY modules
 ILB ----- Intermediate Language Block
 IPSC ----- Information Processing Standards for Computers
 ISP ----- Indexed-Sequential Processor
 I/O ----- Input/Output
 JCS ----- Joint Chiefs of Staff
 JDA ----- Joint Deployment Agency; now a part of USTRANSCOM
 JDS ----- Joint Deployment System
 JDSSC ----- Joint Data Systems Support Center; now named DSSO
 JMH ----- Joint Mission Hardware
 JN ----- NMCS ADP Directorate of DCA
 JNG ----- General Applications Division of JN
 JNGG ----- Information Systems Branch of JNG; the OPR for MAGIC
 JOPES ----- Joint Operation Planning and Execution System
 J-3 ----- Directorate for Operations for the Joint Staff
 MAGIC ----- Mapping and Graphic Information Capability
 Mitre ----- The Mitre Corporation; non-profit corporation spun off from
 MIT to support analyses for the Government
 MME ----- Master Mode Entry
 MTXGEN ----- The GIPSY Matrix Generation Module
 NMCC ----- National Military Command Center
 NMCS ----- National Military Command System
 NMCSSC ----- National Military Command System Support Center (forerunner
 of NMCC)
 Object Code ----- Native machine instructions as generated by a
 compiler/assembler
 OPR ----- Office of Primary Responsibility
 PCS ----- Process Control Statement

PICFIL ----- The GIPSY Picture File
 PLACES ----- The GIPSY PLACES File
 PLOT10 ----- A terminal-based graphics package developed and fielded by
 Tektronix, Inc
 PROCLIB ----- The GIPSY Process Library
 QDF ----- Qualified Data File
 QDT ----- Qualified Data Table
 Q* ----- A Honeywell relocatable system load file used by GIPSY to
 hold a default subroutine library
 RDBMS ----- Relational Database Management System
 SQL ----- Structured Query Language; a standardized 4GL RDBMS language
 SYNTAX ----- The GIPSY Syntax Module
 SYSDEF ----- The System Definition File
 Tektronix ----- Tektronix, Inc. of Beaverton, Oregon; manufacturers of
 graphic workstations and developers of graphic software
 (i.e., PLOT10)
 TM ----- Technical Memorandum
 TRMDEF ----- The Terminal Definition File
 TSR ----- Technical Support Request
 TSS ----- Time Sharing System
 TS8 ----- Virtual Time Sharing System
 TSSGIP ----- The GIPSY Kernel Program for TSS
 TTY ----- Teletype
 UM ----- Users Manual
 USAA ----- The USA Map File
 USERID ----- Uniquely identifies a particular user already known to the
 H6000/DPS8
 VIP ----- Visual Information Projector
 V* ----- Honeywell Visual Display System Console File
 WIN ----- WWMCCS Information Network
 WIS ----- WWMCCS Information System
 WITS ----- WSGT Intelligent Terminal System
 WORLD ----- The WORLD Map File that contains much less map data than
 WORLD2
 WORLD2 ----- The WORLD2 Map File created from the CIA World Data Bank II
 map file
 WSGT ----- WWMCCS Standard Graphic Terminal; an Aydin 5807 graphics
 workstation
 WWMCCS ----- Worldwide Military Command and Control System
 WWS ----- WIS Early Products Workstation

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 2. SYSTEM SUMMARY

This section provides a general description in non-ADP terminology of current GIPSY and the major requirements of the proposed system--MAGIC.

2.1 Background

The following subparagraphs provide background information concerning the uses and purposes for the current systems (GIPSY and GIPSYmate) and the proposed system (MAGIC).

2.1.1 Current System. The following subparagraphs provide pertinent background information to the current systems in place

2.1.1.1 Graphic Information Presentation System (GIPSY). GIPSY provides a general purpose graphics and information display capability. It combines the tools of data retrieval, information processing, and formatted reports using tabular, graphic, and geographic displays into a single, integrated, on-line interactive system for use in briefings, presentations, and reports.

GIPSY was designed in 1976 as an outgrowth of the effort to satisfy the graphic needs of the National Military Command System (NMCS). Although originally oriented toward the NMCS user, GIPSY is equally useful to other WWMCCS sites and, as a result, GIPSY was accepted as the WWMCCS standard graphics software in 1978.

The initial capability requirements used in GIPSY's early development were extracted from a variety of technical documents which included Technical Support Requirements (TSRs), requirements studies, and related documents. These documents were authored by an equally diverse group of sources--the Command and Control Technical Center (CCTC), Mitre Corporation, the National Military Command System (NMCS), the National Military Command System Support Center (NMCSSC), and the Operations Directorate for the Joint Staff (J-3)--exhibiting the widespread graphics needs that GIPSY now satisfies. The original baseline system requirements mandated both a basic user, and an analyst-oriented graphics capability which could display data extracted from a variety of data sources in both graphic and alphanumeric form.

Over the past 13 years, GIPSY has grown and expanded in both size and capabilities while providing an effective interactive graphics system geared to an audience of users at least one step removed from the programmer. GIPSY provides a set of operational tools that allows the applications analyst to build a user-friendly interface between the staff-level users and their data systems. Thus, GIPSY functionality may be tailored to the specific information it needs to display and to the users the system must serve.

Currently, GIPSY has in excess of 1,000 separate but interrelated programs, nearly 200 common data structures, and a variety of supporting data files and system utilities. Written mostly in FORTRAN 66, the overall design of the

system incorporates a device-independent approach based on the Core Graphics System which supports multiple dissimilar graphic output devices. Even so, GIPSY was written to maximize the PLOT10 capabilities Tektronix 4014-1 graphics terminal and most terminal output is displayed in Tektronix format. Honeywell assembler coding (GMAP) accounts for approximately 13 percent of the total system and is concentrated in programs concerned with internal processing and some input/output (I/O) functions.

2.1.1.2 GIPSYmate. On 5 October 1984, the Common User Contract (CUC) was awarded to the IBM Corporation. As a result, IBM developed the Worldwide Military Command and Control System (WWMCCS) Information System (WIS) Workstation (WWS) Early Product which is based on the IBM Personal Computer (PC/XT). The WWS Early Product has a resolution of 720 X 350 pixels and is configured with 640 kilobytes (Kb) of internal memory, two removable 5 megabyte (Mb) hard disks, and one 5.25 inch floppy disk drive capable of storing 360 Kb. It is supplied with commercially available business graphics software (Energraphics) which produces bar, pie, and line charts, but will not satisfy the graphics requirements of the WWMCCS community. The WIS JPMO Letter of Technical Support Requirements tasked the JDSSC to design and develop an interim graphics interface between the WWS Early Product and the H6000-based GIPSY. This interface, developed for the eight color version WWS Early Product, provides the WWS Early Product user with access to all GIPSY capabilities via the Honeywell Visual Information Processor (VIP) 7705 bisynchronous protocol.

The Initial Operational Capability (IOC) of the graphics interface was successfully demonstrated to the WIS Joint Program Manager (JPM) on 19 March 1986 and released to the user community as GIPSYmate 1.0, August 1986. As a result of the demonstration, the WIS JPMO tasked the JDSSC to remove certain limitations present in the IOC as well as to provide the WWS Early Product user community with several capabilities that were available for the now defunct WWMCCS Standard Graphics Terminal (WSGT) through the WSGT Intelligent Terminal System (WITS) Briefing Aids subsystem. These capabilities were integrated into GIPSYmate and made available to the users with GIPSYmate Release 2.0, April 1987.

Support of an additional VIP emulator, the Enhanced Terminal Capability (ETC), produced by ECDSC, USEUCOM, was integrated into GIPSYmate and made available to the users with GIPSYmate Release 3.0, August 1988. Enhancements of a DOS window and an impending alarm timeout notification were added to GIPSYmate and made available with GIPSYmate Release 3.1, April 1989.

The Joint Chiefs of Staff (JCS) have tasked the Worldwide Military Command and Control System (WWMCCS) Information System (WIS) Joint Program Management Office (JPMO) with the development of the Joint Operation Planning and Execution System (JOPEs). The development of JOPEs requires the support of a graphics software system. The WIS JPMO plans to utilize GIPSY and GIPSYmate to provide the graphics support for JOPEs. In order for GIPSYmate to function in the JOPEs environment and exploit the capabilities of the target workstation for JOPEs implementation, the GIPSYmate system needs to be

modernized using Ada as the high order language and implementing the Graphical Kernel System (GKS).

Currently, the WIS Joint Project Management Office (JPMO) has been dissolved and Defense Systems Support Organization (DSSO) has been designated to continue the functions of WIS JPMO. Due to this change, the program has been renamed to WWMCCS ADP Modernization (WAM), thus the procuring workstation is referred to as the WAM Workstation (WWS).

2.1.2 Proposed System. The MAGIC effort has evolved from and will build upon the modernization of the WWMCCS standard host-based Graphic Information Presentation System (GIPSY) and the Z-248 PC-based modernized GIPSYmate system. Designed and developed to meet the needs of a new generation of WWMCCS users, MAGIC will be fielded as a resident system on a workstation and will present a menu-based graphical user interface (GUI) to the user that integrates (as transparently as feasible) the COTS packages also resident on that platform. Processing facilities appropriate to both sophisticated and novice users will be supported as well as the ability to access the full range of data base types found on the WWMCCS host. Functionally, the MAGIC user will have the capability to perform data retrieval and manipulation operations, business graphics displays, geographic and geodetic mapping displays, slide show generation, and graphic editing operations.

Specifically, MAGIC will utilize the workstation-resident Oracle COTS package for its DBMS-related operations and data files on both the workstation and an H6000 host may be accessed. Furthermore, MAGIC operations related to data management may be accomplished through any one of three methods: (1) Assisted SQL which is the default menu-driven method, (2) Unassisted SQL which bypasses the menus and permits the free-form entry of sophisticated SQL queries, and (3) PCS Mode which also bypasses the menus but accepts GIPSY language PCS statements for uploading to the H6000 host for remote processing.

MAGIC's business graphics capability will be provided through the Wingz product available from Informix, Inc. MAGIC's geographic and geodetic mapping functions will be provided through an additional COTS package--the DeLorme Mapping System. These and other COTS packages resident on the workstation will be integrated through MAGIC's GUI providing the WWMCCS user an incomparable value--the power and compatibility of COTS linked together and made available through a single menu style that is user-friendly, flexible, and doesn't require manual traversal between COTS packages.

While the bulk of MAGIC processing will be accomplished on the workstation, some communications and data retrieval/manipulation software will be utilized on the H6000 host.

2.2 Objectives

The primary objectives of MAGIC are to meet the goals of software engineering, ease portability to the WWMCCS Workstation (WWS), incorporate industry and American National Standards Institute (ANSI) standardization of graphics

processing, and maintain the functionality and user-friendliness of GIPSY and GIPSYmate. The above objectives are expanded upon below:

- a. Strive to meet the goals of software engineering:
 - (1) Modifiability, meaning that small changes can be made without altering the design structure and changes in one part of MAGIC will not adversely affect unrelated parts of the system.
 - (2) Efficiency, which indicates that available resources are used in an optimal manner.
 - (3) Reliability of MAGIC will be taken into account in the design to ensure that circumstances within the developer's control are handled. MAGIC will be at least as reliable as current GIPSY.
 - (4) Understandability will be achieved by utilizing a High Order Language (HOL) which allows the use of meaningful names, has no length restriction on names, and has selected component notation for traceability of entities.
- b. Use of an existing HOL will ease portability of MAGIC to various hardware platforms.
- c. Incorporate industry- and ANSI-standardization of graphics processing in the design of MAGIC:
 - (1) The older methods of developing graphics systems with device-specific graphics processing will be replaced by a standardized implementation of X Windows and OSF/Motif.
 - (2) A Motif-compliant graphical user interface (GUI) will be used to provide an industry-standard, workstation-based front-end for MAGIC's user interaction.
 - (3) Commercial Off-The-Shelf (COTS) packages will be used wherever possible to provide enhanced graphics capabilities.
 - (4) X Windows will be used to provide network transparency and portability among other workstation-based platforms.
- d. Maintain the user-friendliness and functionality of GIPSY and GIPSYmate.

2.3 Existing Methods and Procedures

GIPSY provides a general purpose graphics capability that operates on the Worldwide Military Command and Control System (WMMCCS) computer, a Honeywell Information Systems (HIS) 6080/DPS8000. The system supports the data

presentation needs of the customers of the DSSO, which includes all of the WWMCCS sites.

2.3.1 Capabilities of the System. GIPSY is a general-purpose graphics and information presentation system. It combines the tools of data retrieval, information processing, and tabular, formatted, graphic, and geographic reports into a single, integrated, on-line interactive system. It is a file- and data-independent system that is driven by a high-level, user-oriented language. The graphic capabilities are implemented using a device-independent approach that allows the single, integrated system to support multiple dissimilar devices.

Although the graphic reports capability was the primary basis for the initiation of the system, GIPSY very effectively serves as an information handling system to connect the user's data base to a large set of on-line, interactive query and report capabilities. GIPSY may even be run in the batch environment to support requirements involving high-volume input/output.

2.3.2 Structure of the System. GIPSY consists of 10 separately loadable modules which accomplish system initiation, syntax processing, data selection, matrix generation, statistical graphics, plotter interface, geographic and formatted report processing, as well as user calls processing and batch processing. The system organization is shown in figure 2-1.

2.3.2.1 System Initiation. Each module is stored on a separate program link file (H*). Processing is controlled by the Executive Module (EXEC), which sets up the module execution sequence at system initiation time. Figure 2-2 shows the Executive Module and the primary control files. By having separately loadable modules, resources are conserved by keeping in memory only those programs necessary for the task at hand. All modules are supported by internal and external data structures and control processes.

The start-up of GIPSY parallels that of other Time Sharing Systems (TSS), such as EDIT or FORT. The user executes the system by entering the command "GIPSY". The GIPSY Kernel for Time Sharing (TSSGIP), which is resident in the TSS Command Library (CMDLIB), performs the necessary functions to initiate the GIPSY Executive Module and passes the system specifications and first line of the run options to the GIPSY Executive.

Once initiated, the GIPSY Executive Module performs several key functions. First, the station code to which the user is logged on is fetched from the system, and the attributes of the specific terminal being used are extracted from the System Definition file (SYSDEF). Then, the Executive validates the specified run options, performs the required processing to satisfy all options specified, and sets up the execution sequence of the modules needed for the current session. Control is then transferred to the next GIPSY module.

2.3.2.2 Syntax Processing. The Syntax Module (SYNTAX), shown in figure 2-3, is normally the second module to be executed. This module accepts the input, output, and processing specifications from the user in the form of standard

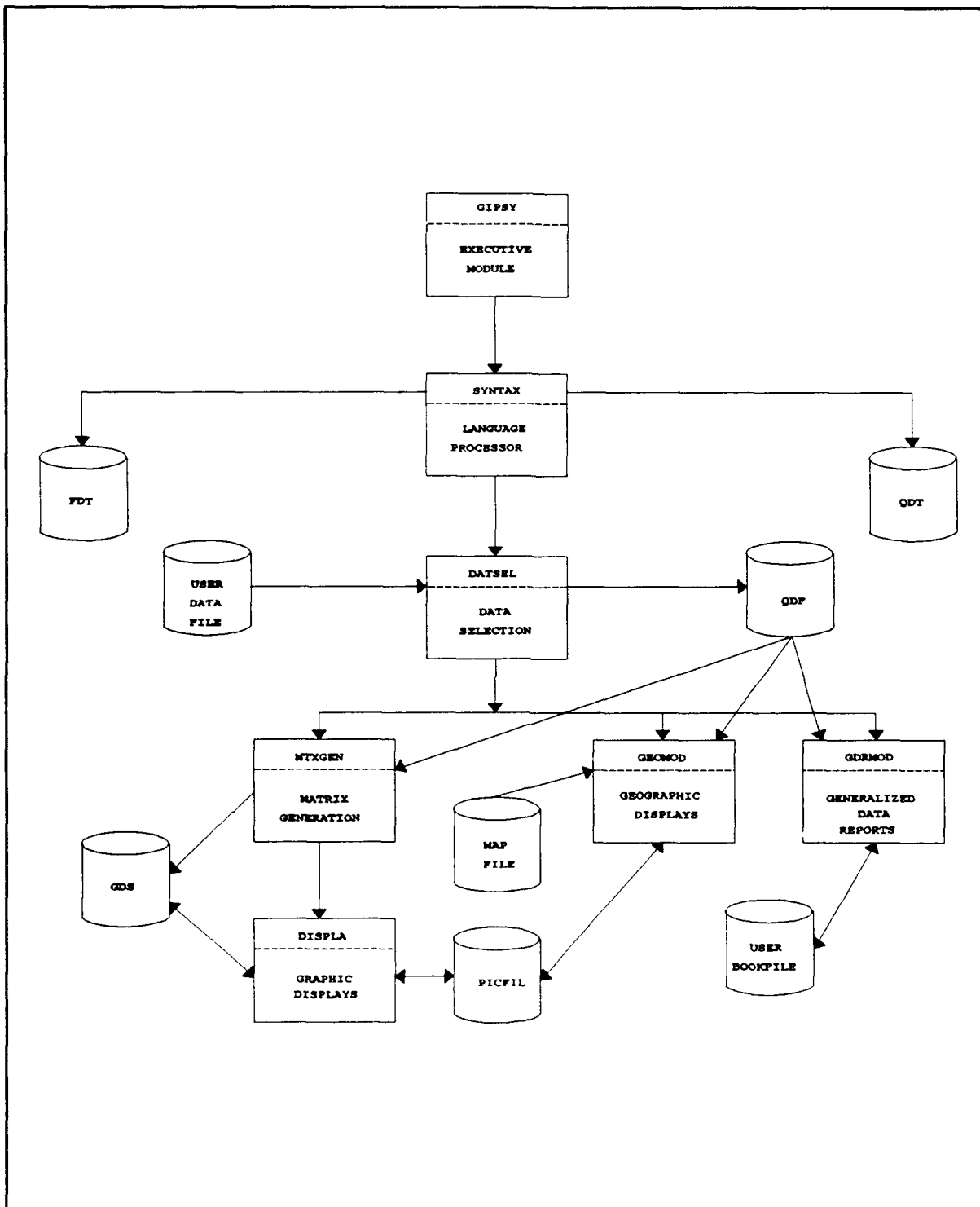


Figure 2-1. GIPSY's System Organization

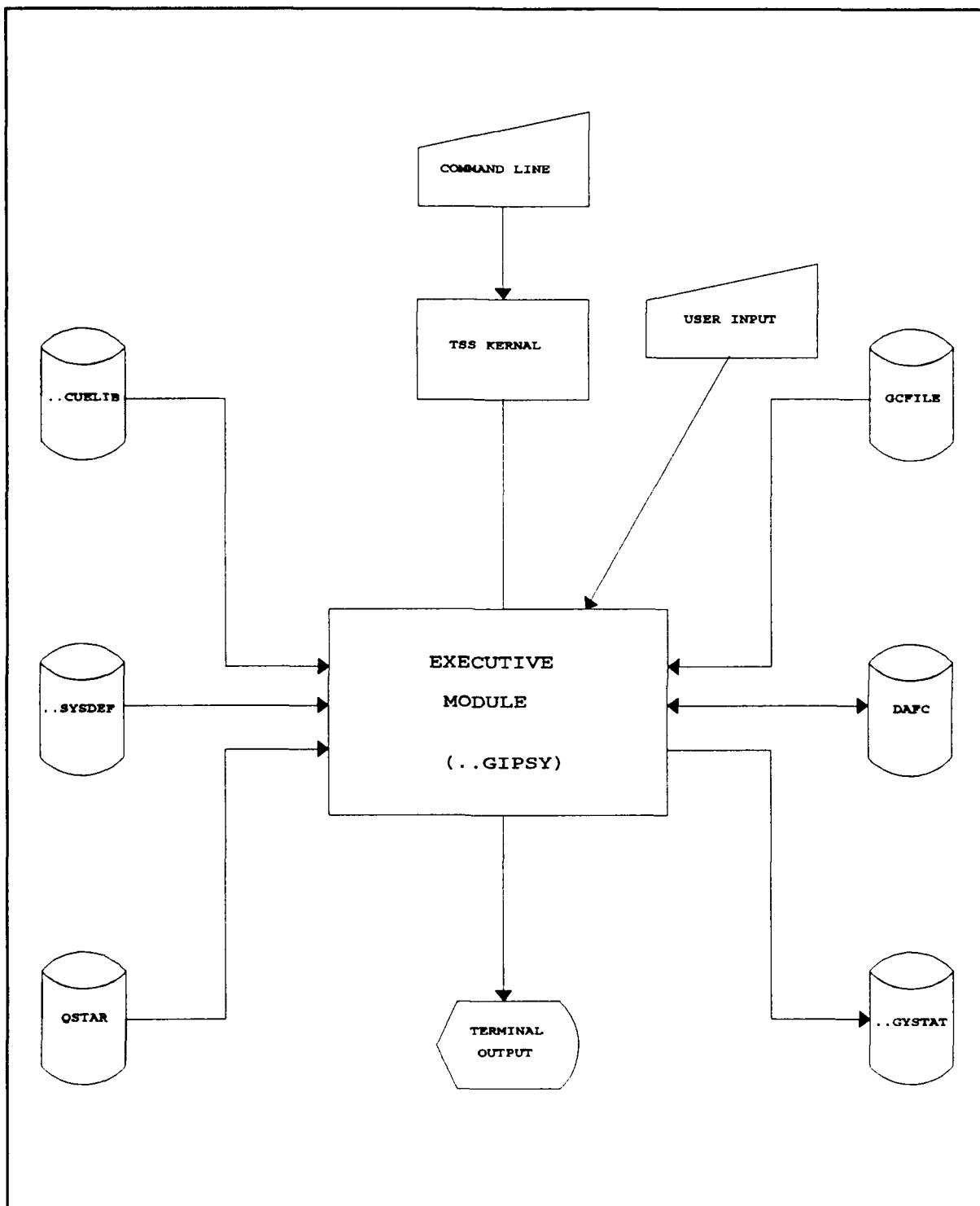


Figure 2-2. Executive Module

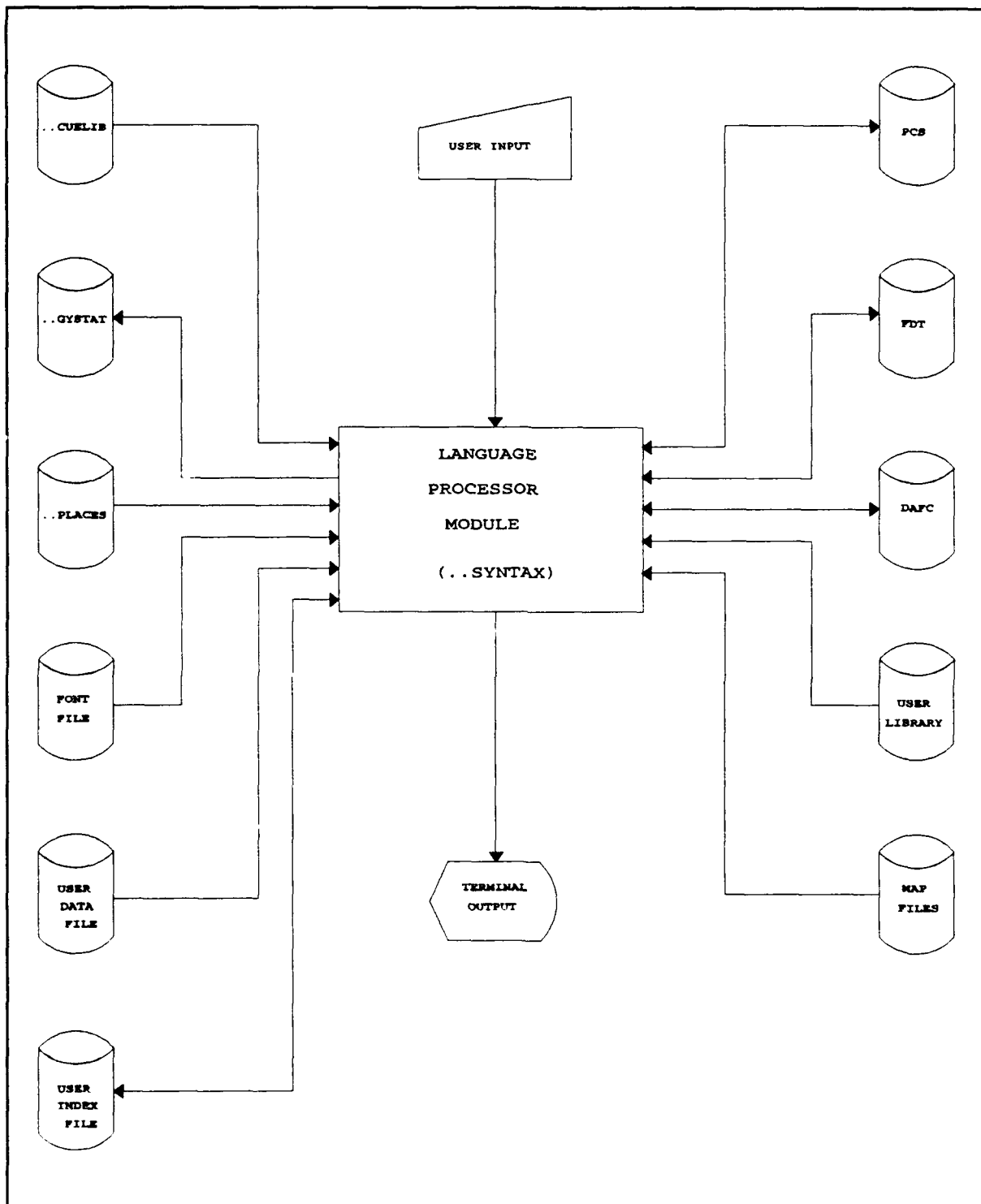


Figure 2-3. Syntax Module

GIPSY commands. The user specifications are immediately examined for both syntactic and semantic errors. If an error is detected, the user is given the opportunity to make the necessary corrections. This interactive process provides the user with a conversational capability in that GIPSY not only provides for error detection and correction but also informs the user of what is expected as input at any point in the process. In addition to performing the validation of commands, the syntax process involves the generation of intermediate information and control structures that are used in subsequent processing. These intermediate language block (ILB) structures are then converted to executable object code, which is relocatable. This process provides the efficiency needed to perform many calculations and qualifications by negating the need for an interpretive process that would otherwise be required to perform these functions.

2.3.2.3 Data Selection. The Data Selection Module (DATSEL), shown in figure 2-4, is the third step in the standard process and is the key interface between the user's data file and the data presentation features of GIPSY. The data selection process involves reading the user's data following the File Descriptor Table (FDT) format, then writing out the desired subset to the Qualified Data File (QDF). During this process data modification is performed, records are retrieved based upon retrieval criteria, and records are sorted based upon selected sort keys. The resulting QDF is passed on to be used as input to the report building process.

2.3.2.4 Matrix Generation. The Matrix Generation Module (MTXGEN), shown in figure 2-5, is the next step when producing statistical reports. It involves the further reduction of the data into a form suitable for both tabular reports and statistical graphs. This is accomplished by consolidating the user data into a concise Graphic Data Set (GDS) that contains all information necessary to produce the report. The method used in constructing the GDS is dictated by the tabular report options specified during the syntax process. A set of processing algorithms provides the flexibility and efficiency required to handle the numerous permutations possible in tabular report specifications.

2.3.2.5 Statistical Reports Processing. The Statistical Reports Module (DISPLA), shown in figure 2-6, is executed as the final step to generate statistical alphanumeric and graphic reports. The displays are created using the GDS matrix as the primary input to the process. The type of report (e.g., tabular report, line graph, bar chart, pie chart) is controlled through the use of display specification commands. The internal operation of the display process automatically tailors the report to the device being used. All associated functions, such as scaling, shading, legends, character selection, and paging, are dynamically determined based upon the data and device characteristics of the terminal being used. In addition to report generation, a powerful report transformation capability is available for interactively modifying the original GDS structure. This capability provides the user with the ability to dynamically restructure the data based upon a set of report modification specifications.

2.3.2.6 Geographic Processing. The Geographic Reports Module (GEOMOD), shown in figure 2-7, is executed when producing geographic reports. The geographic

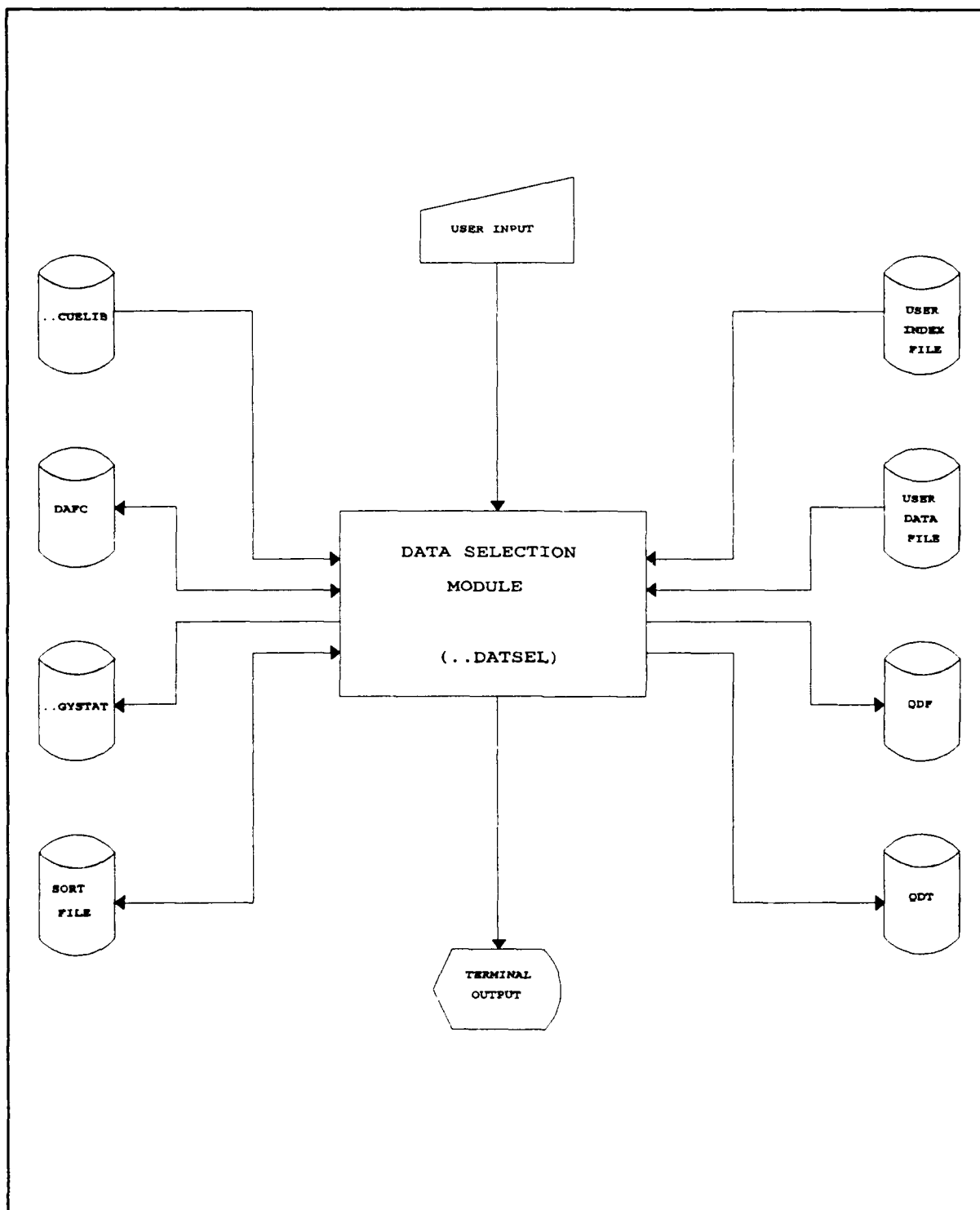


Figure 2-4. Data Selection Module

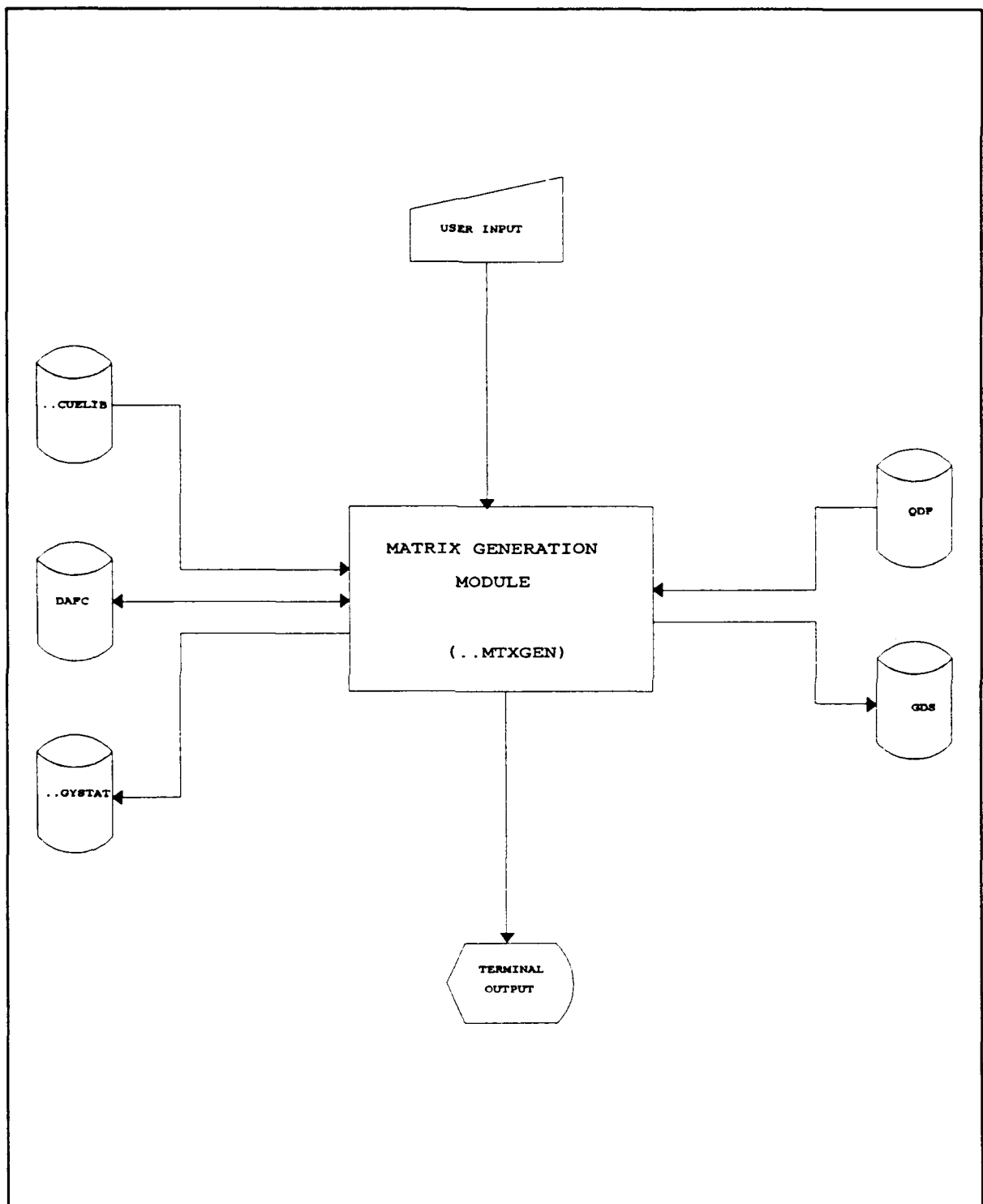


Figure 2-5. Matrix Generation Module

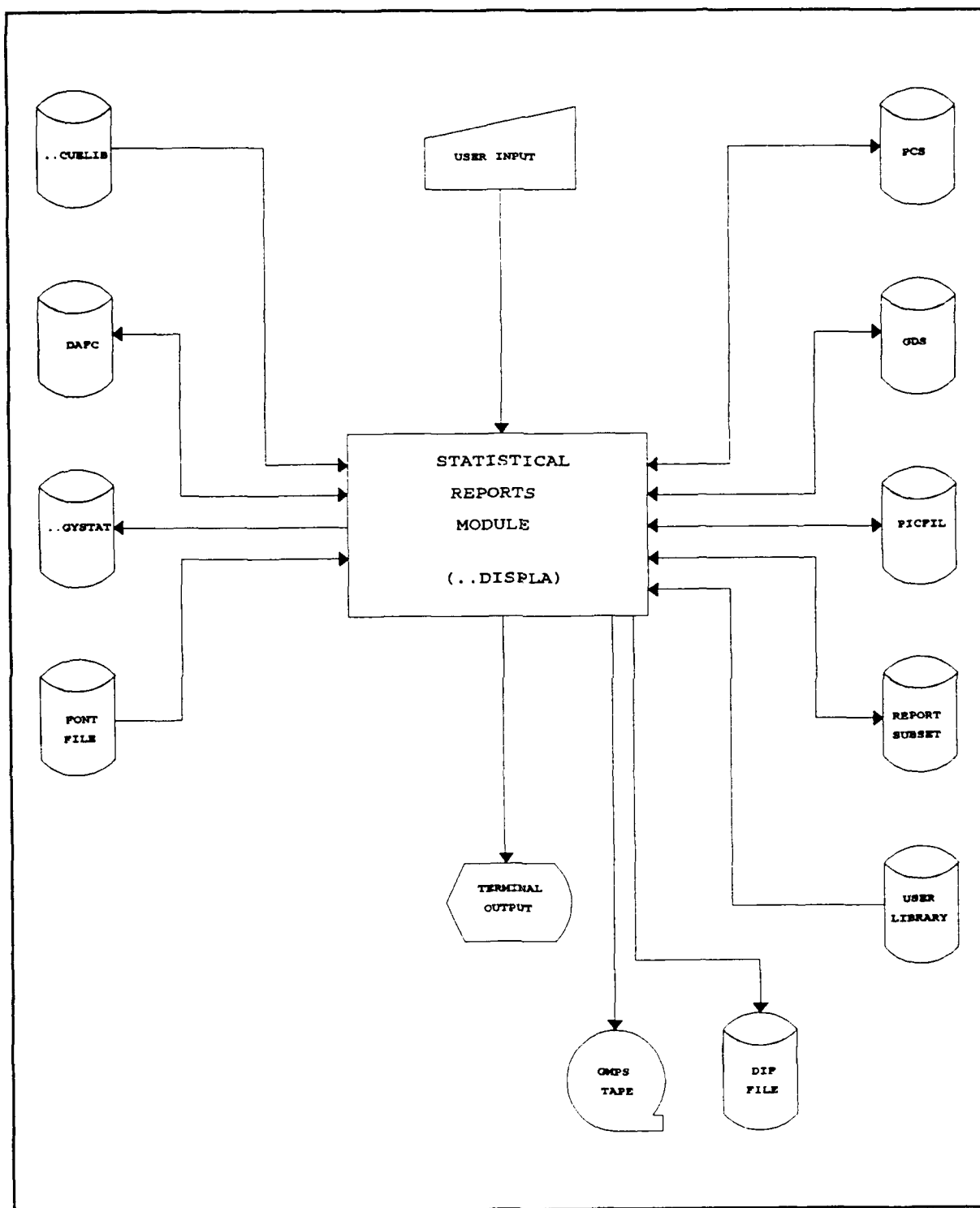


Figure 2-6. Statistical Reports Module

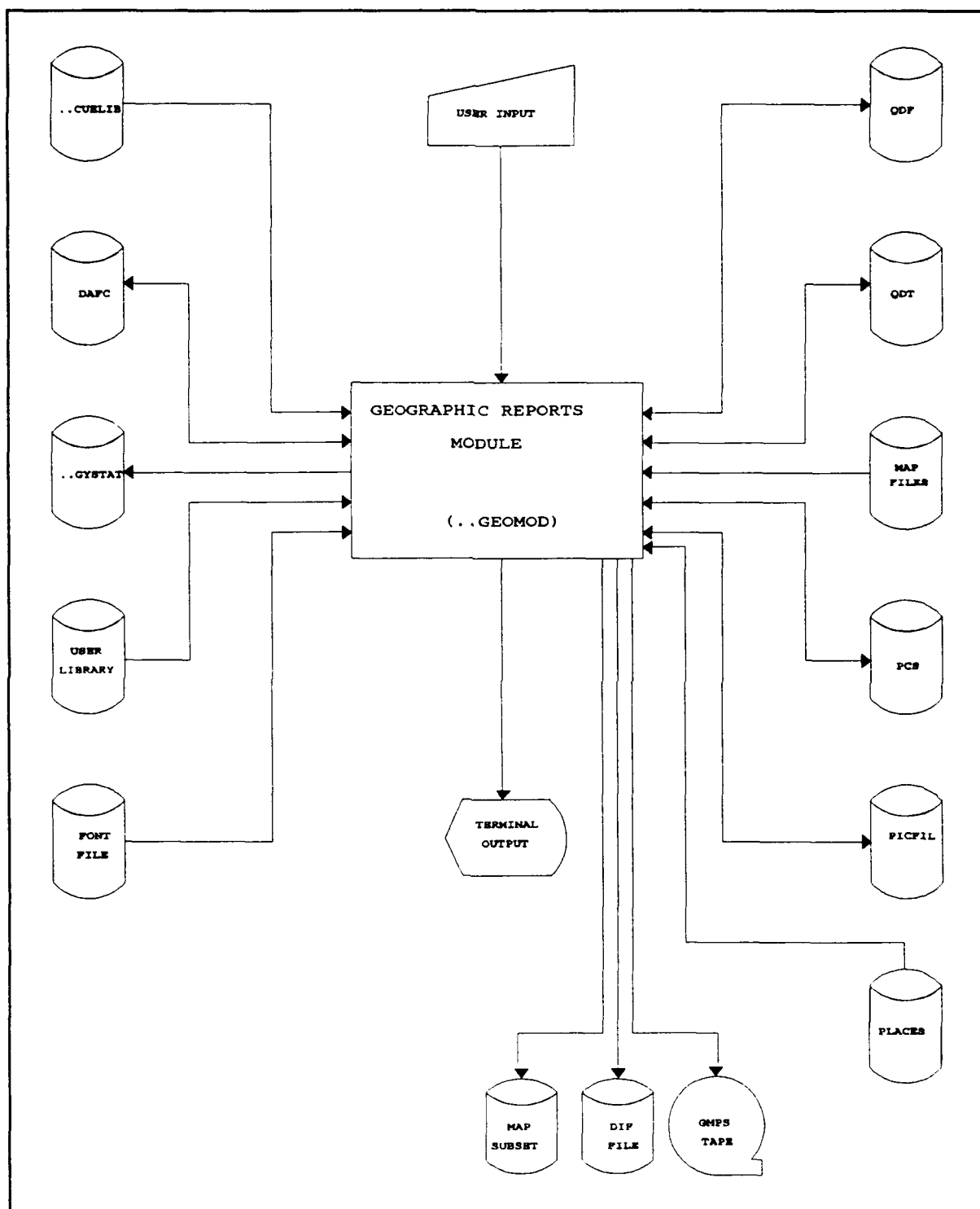


Figure 2-7. Geographic Reports Module

displays are created using a combination of user-supplied data and a map file. The user-supplied data is used in the generation of geographic symbol and track plots. The map file contains the necessary information to display the geographic boundaries. The internal operations of the system provide for the combination of user data and map file data into the final geographic report.

Associated functions such as clipping and scaling are automatically provided by the module processing operations. The geographic module interacts directly against the data in the QDF. This data can be further subset interactively, thereby allowing the user to qualify which data is to be displayed. A print capability is also available to list the contents of the qualified records onto the terminal in the form of an alphanumeric listing.

2.3.2.7 Formatted Reports Processing. The Generalized Data Reports Module (GDRMOD), shown in figure 2-8, allows for complex report formatting. Formatted reports are generated by using a combination of user-supplied data and user-specified formatting details. The internal operations of the system allow the user data to be transformed and directed into user BOOKFILES. These BOOKFILES may be displayed on the terminal, directed to a printer, or stored on a disk file. A paging feature allows the user to go directly to a specific section of the report and then page forward or backwards.

2.3.2.8 User Calls Processing. The Call Module (CALMOD), shown in figure 2-9, is used to execute stand-alone subroutine calls which are actually Honeywell relocatable system load files (Q*). The usefulness of this module arises from the fact that it occupies less memory than SYNTAX, DISPLA, GEOMOD, or GDRMOD, thereby resulting in faster execution of user calls. The module is entered via the TRANSFER command.

2.3.2.9 Batch Processing. GIPSY is executed in the batch environment by using the module BGIPSY. This module is loaded into core memory as a front-end driver to the GIPSY TSS-based modules. One of the most important functions of this driver is the translation of TSS DRL commands into batch-based Master Mode Entry (MME) commands. The BGIPSY module is shown in figure 2-10.

2.3.2.10 Related Structures. The following external data structures support the execution of the GIPSY modules as shown in figures 2-2 through 2-10.

- a. BOOKFILE. Holds the BOOK (formatted report) created by the GDRMOD module. The file is temporary (file code 25) by default, but may be replaced by a user-specified permanent file.
- b. CUELIB. The Cue Library contains the error and expected messages used by GIPSY. It also contains file pointers to the modules and data files accessed during GIPSY execution. It is a permanent file which is part of the GIPSY release.
- c. DAFC. The Directive Action File Control serves as a disk storage area common to all modules. During module transfer, all essential

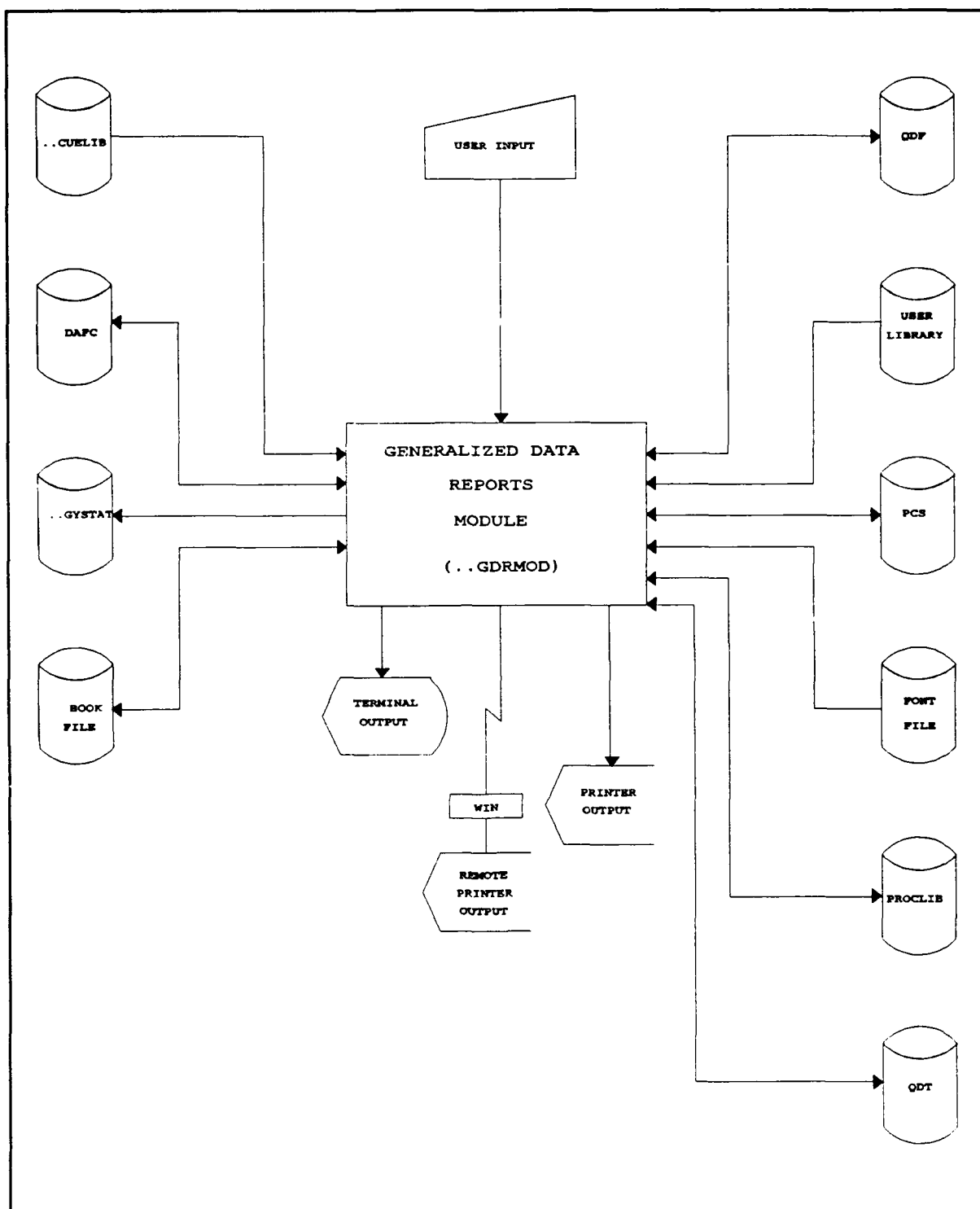


Figure 2-8. Generalized Data Reports Module

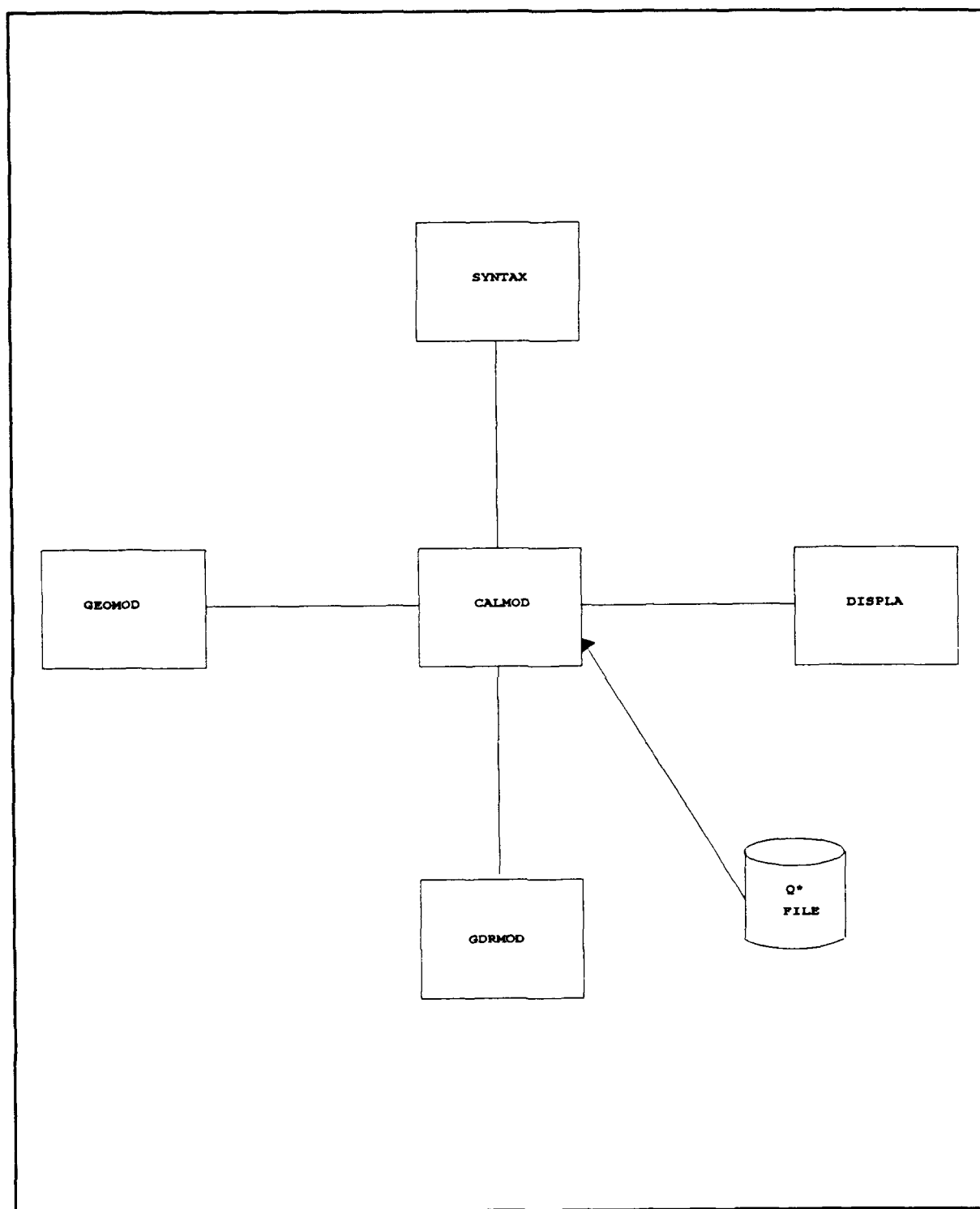


Figure 2-9. Call Module

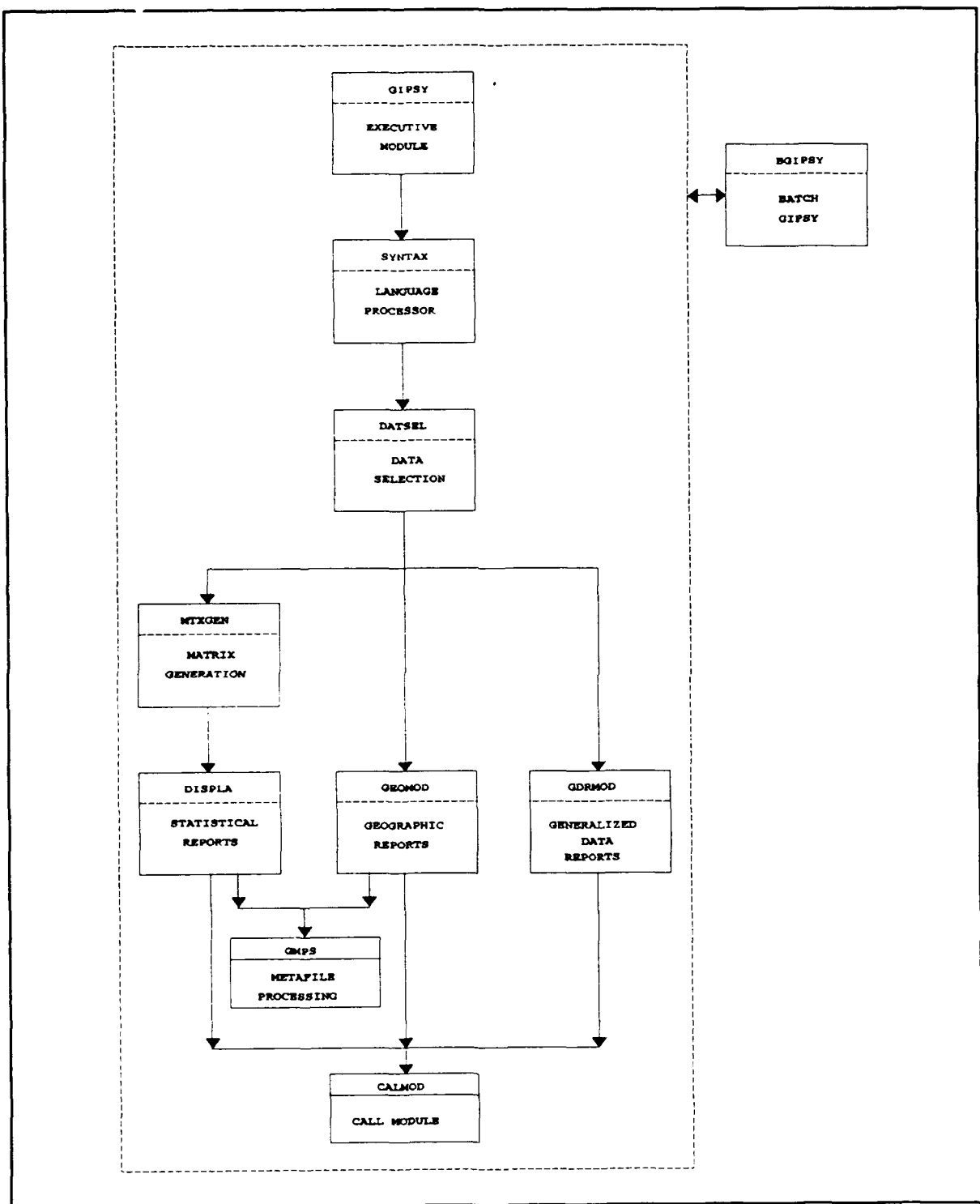


Figure 2-10. Batch GIPSY Module

common areas and end-of-memory (EOM) arrays are passed via the DAFC. The DAFC is a temporary file (file code 21), but may be saved to a permanent file. The permanent DAFC may be used as input for a new GIPSY session.

- d. DIF FILE. The Data Interchange Format file is used to support standardized data exchange between applications programs. GIPSY has the capability to produce DIF files for both QDF and GDS files. These files may then be ported as input to other computer systems.
- e. FDT. The File Descriptor Table is a sequential file created by GIPSY which defines the fields in the user data file, as well as work area fields. The structure is normally stored internally, but may be saved to a permanent file.
- f. FONT FILE. Contains one or more alternate character sets for use on the WIS Workstation (Early Products Workstation). Normally, this is a permanent file that is part of a GIPSY system release but it can be user-created.
- g. GCFILE. The GIPSY Command File contains command line options which can be stored. This option is used when there is not enough room for all of the desired options to be specified on a single line of input.
- h. GDS. The Graphic Data Set contains the matrix data requested by the user in the Build Tabular Report or Build New Report structure. It is the data input to the DISPLA module. Normally a temporary file (file code 07), it can be saved to permanent space.
- i. GYSTAT. The GIPSY Statistics File is part of the GIPSY release which stores statistics on GIPSY usage. Items monitored include USERID, terminal ID, core memory used, processing time, and number of I/Os.
- j. MAP FILES. The GIPSY release contains three different map files: WORLD2, WORLD, and USAA. Each varies in degree of detail, area of coverage, and type of political and topographical information available.
- k. MAP SUBSET. This is a user-specified permanent file to which GIPSY can output a portion of the WORLD2 file in the form of geographic points, track numbers, and track types.
- l. PCS. The Process Control Statement file is a sequential file which contains GIPSY commands. The PCS can be a user-created and then input to GIPSY, or the commands entered into GIPSY can be saved off into a PCS file.
- m. PICFIL. The Picture File contains the moves and draws, textual data and supporting information necessary to recreate a graphic report on the terminal or a plotter. The default is a temporary file (file

code 08), however, it may be saved onto permanent space (file code 06).

- n. PLACES. The Places File contains a cross-reference between place and location names and the geographic coordinates that describe the area. It is a permanent random file sent out with the GIPSY release.
- o. PROCLIB. The Process Library is a file in DAFC format which contains a set of GIPSY processes. This file is saved by GIPSY onto a user-specified file, which can later be accessed from within the GDRMOD module.
- p. Q*. The Q* is a Honeywell relocatable system load file whose contents are searched for required system routines prior to using the actual system routines of the same name. As such, this is GIPSY's default subroutine library which contains about 20 commonly used subroutines and allows GIPSY to function as a completely independent TSS subsystem. These subroutines are used in Field Tables, Output Tables, and stand-alone calls. Each GIPSY release includes its own Q* default library.
- q. QDF. The Qualified Data File contains the output records of the data selection process. The file is temporary (file code 10) by default, however, it can be saved and accessed later, or the user may specify a permanent file prior to data selection.
- r. QDT. The Qualified Data Table defines the data fields in the QDF. This structure normally resides internal to GIPSY, however, it may be saved to and recalled from a permanent file.
- s. REPORT SUBSET. This is a temporary file (file code 03), used in the SUBSET function of the DISPLA module wherein a GDS is subset by a user.
- t. SORT FILE. The sort file is a temporary file (file code 03) in which retrieved data is sorted prior to being written to the QDF.
- u. SYSDEF. The System Definition file contains the attributes associated with the various terminal types plus a cross-reference between station codes and terminal types. This file is part of the GIPSY release and is created from the Environment (ENVIRO) and Terminal Definition (TRMDEF) files.
- v. USER DATA FILE. This is any data source from which GIPSY retrieves data which is subsequently used to produce the desired report. The file is entirely user-controlled and can be in sequential or Indexed-Sequential Processor (ISP) format.

- w. USER INDEX FILE. This is a user-created file, in the format required by GIPSY, which is used as an index in retrieving against large ISP files. This is not the ISP index.
- x. USER LIBRARY. This is a library of user-written subroutines assembled into Q* format for input to GIPSY.

2.3.3 System Specification. GIPSY operates on the Honeywell Information System (HIS) 6080 computer under the virtual General Comprehensive Operating Supervisor (GCOS 8) in accommodation mode. The system has been programmed and tested to operate on the Honeywell 6080. GIPSY functions as a subsystem under the Time Sharing System (TSS). It executes within 16,000-35,000 words (93-205 Kb) of memory. GIPSY can operate on all models in the 6000 Series that have remote processing capabilities. The non-graphic features (e.g., tabular and printed reports) are compatible with any ASCII terminal, such as the Honeywell 7705W VIP; WIS Workstation (Early Products Workstation); and the Tektronix 4014-1 graphics terminal. The graphic features (e.g., graphs and maps) require a graphics terminal such as the Early Products Workstation (IBM 3270 PC/XT) or the 4014-1. Figure 2-11 illustrates the system configuration used at the DSSO.

GIPSY uses only those GCOS 8 routines and services that are normally available to slave programs. The existing system has not been restricted to any one level of GCOS and should operate with any subsequent GCOS 8 release. It is programmed in FORTRAN and GMAP, with the GMAP programs constituting approximately 13 percent of the system. In addition to its own applications program, the following HIS 6000 software must be included to operate and maintain the current GIPSY system:

- a. GCOS 8 (WWMCCS version)
- b. Time-Sharing System (TSS)
- c. Datanet software (WWMCCS version)
- d. GMAP Assembler
- e. FORTRAN 66 Compiler.

GIPSY parallels TSS in its organization in that major functions in GIPSY are independently executable modules. This method of implementation allows each major component of the system to be detachable and hence, replaceable or augmentable by any user function that can communicate with GIPSY.

Each user of GIPSY gets a separate copy of the system module being executed but, by virtue of operating in the TSS region of memory, each user may be swapped out either when idle or when a higher priority request comes in. Since an interactive system is idle most of the time, the cost of giving each user a copy of GIPSY is minimized.

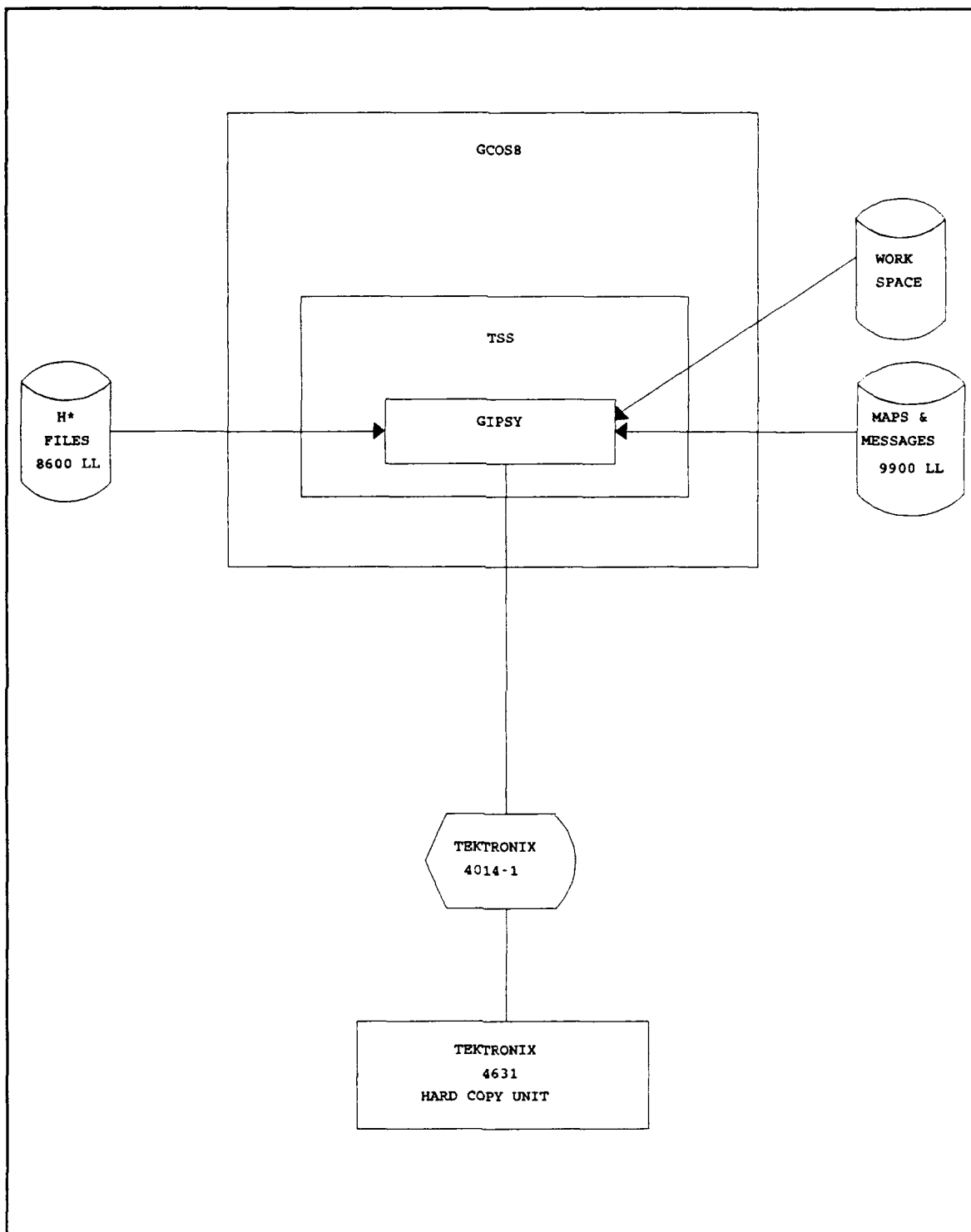


Figure 2-11. GIPSY's System Configuration at the DSSO

GIPSY requires no special permissions, priority, privacy, or considerations from the operating environment. The system is designed to allow use of a database as it currently exists. There is no provision for creating or modifying the file, either deliberately or inadvertently, during a GIPSY session. The user is thus assured that the data input file is protected at all times.

2.3.4 Deficiencies and Limitations. As GIPSY has evolved over the past 13 years, much of its functionality has been added quickly in response to user requirements. While responsive to user needs, some of these additions have created problems such as incomplete system documentation (i.e., the lack of a Functional Description, System/Subsystem Specification, Software Unit Specification, or Test Plan). In addition, the inherent inadequacies of the FORTRAN compiler, in conjunction with the many enhancements, have created difficulties in maintaining and/or enhancing GIPSY in a cost-effective manner. Also, the computer industry itself has been subject to evolutionary changes since GIPSY was developed.

Some of the inconsistencies and shortcomings within GIPSY that affect the system today and will present challenges for future development are listed below:

- a. GIPSY's hierarchical structure has grown less distinct over the years:
 - (1) GIPSY is difficult to modify:
 - (a) Small changes may alter the design structure.
 - (b) Changes in one part of GIPSY may adversely affect other parts of the system because of the lack of traceability and the close interrelationships of the various modules.
 - (2) GIPSY contains duplicate and unused code:
 - (a) The design structure is often unclear to the maintenance programmers.
 - (b) Changes often lead to duplicate code and sections of code that are no longer executed.
 - (3) GIPSY is difficult to understand:
 - (a) GIPSY's code is often obscure.
 - (b) Data and variable names are often meaningless and are not documented.
 - (c) The overall design for some sections of code is not easy to obtain or derive.

- (4) GIPSY is expensive to modify:
 - (a) No one individual understands the entire system; many specialists are relied upon.
 - (b) Debugging requires excessive time.
- b. GIPSY does not adhere to modern principles of software engineering:
 - (1) Because of the complexity of GIPSY, programmers often become lost in what a program does.
 - (2) Any routine may call any other routine or common area in the system. There is no way to make parts of the system that should not affect other parts of the system inaccessible.
 - (3) Functionality, such as syntaxing and terminal capability, is spread throughout GIPSY.
 - (4) Logically related resources are not localized. Because massive common areas are utilized, changes are difficult to make, and changing a common area may have a domino effect.
- c. FORTRAN 66 is inadequate:
 - (1) Honeywell FORTRAN 66 is not easily ported to other hardware platforms.
 - (2) The lack of a vigorously-enforced structure within the language allows many coding constructs that are contrary to the basic principles of software engineering.
 - (3) GIPSY handles and manipulates strings via FORTRAN, which is an inefficient language for string manipulation.
- d. GIPSY does not have standard graphics input/output:
 - (1) GIPSY currently contains device-specific code.
 - (2) GIPSY lacks total segmentation capabilities or other standardized graphics processing to permit the interactive modification of portions of graphical displays without requiring complete picture regeneration.
 - (3) GIPSY's metafile (picture file) generation is non-standard and unique to GIPSY itself.

2.4 Proposed Methods and Procedures

This paragraph describes the proposed methods and procedures to be used in MAGIC.

Specifically, MAGIC will be comprised of seven logical subsystems--hereafter referred to as Computer Software Configuration Items (CSCIs)--based upon the functionality they perform and the services they provide both the user and each other. The seven CSCIs are:

- a. Human Interface
- b. Data Management
- c. Business Graphics
- d. Geographic Mapping
- e. Graphic Editor
- f. Slide Show
- g. Internal Processing.

All CSCIs are discussed in greater detail in subparagraph 4.2.3 which delineates their purposes, functions, and how their attendant components will satisfy the requirements provided in appendix A.

In addition to the seven logical CSCIs, MAGIC will incorporate an eighth area of development: Programmer Utilities. While not a fielded CSCI, Programmer Utilities perform important system maintenance and development functions. For a more detailed discussion of these functions, refer to subparagraph 4.2.3.8.

2.4.1 Summary of Improvements. Through the design and fielding of MAGIC, the following improvements will be realized:

- a. A modernized design structure that is clear to staff at different levels (designers, development programmers, maintenance programmers, etc.)
- b. Ease of system modifiability where software maintenance costs are lower, the original design structure is left intact by small changes, and most changes would have to be made in just one place
- c. The use of understandable and readable code which uses both meaningful names and consistent component notation
- d. MAGIC will use the modern principles of software engineering: modularity and localization; abstraction and information hiding; and uniformity, completeness, and confirmability

- e. Enhanced system portability due to the use of a HOL that does not use assembly-level code and hardware-dependent bit manipulations wherever possible. Also the implementation of an X Windows/Motif layer to provide standardized graphics processing
- f. A modern menu-driven system utilizing ANSI and industry standards to provide a GUI that supports both mouse and keyboard access.

2.4.2 Summary of Impacts. MAGIC is anticipated to have the following impacts on the users from the organizational, operational, and developmental standpoints. Cost estimates for these impacts are discussed in section 8.

2.4.2.1 User Organizational Impacts. There are no anticipated organizational impacts.

2.4.2.2 User Operational Impacts. There are a number of likely impacts that a user will realize from an operational standpoint. Most notable among these are:

- a. Lead time required to support new graphic output devices will decrease dramatically. The implementation of X Windows will remove device-dependencies from MAGIC and add network transparency.
- b. The use of Open Software Foundation (OSF)/Motif to provide the GUI will provide MAGIC users with an efficient system based on de facto industry standards for user-friendly interfaces. As a result, the skills learned while operating MAGIC's interface are easily ported to other platforms.
- c. Graphics transferability will be eased. By utilizing Commercial Off-The-Shelf (COTS) packages for graphics generation and modification, MAGIC users will gain portability as well as ease of conversion to other standards.
- d. There will be a possible degradation of system response time depending upon the level of HOL used to replace existing assembly-level code for GIPSY I/O as well as any workstation-based security packages.
- e. While access to the host-based GIPSY will be preserved via MAGIC's interface to it, the MAGIC user of GIPSY services will be using a Structured Query Language (SQL)-based method of access that permits easy loading into an Oracle Relational Database Management System (RDBMS).
- f. Two modes of access will be supported for the MAGIC user--assisted and unassisted. The assisted mode will provide a fully functional GUI with a large number of menu screens, forms, and dialog boxes as well as context-sensitive help screens. The unassisted mode bypasses the GUI and is designed for the advanced user who does not require a

layer of abstraction on top of the underlying COTS packages or, in an experienced GIPSY user's case, translation to GIPSY's PCS language.

2.4.2.3 User Development Impacts. Due to a decision that MAGIC must be developed while the GIPSY and GIPSYmate systems remain supported and in full operation, there are no anticipated user developmental impacts. The existing system will not be withdrawn until such time as MAGIC successfully completes all system integration tests and FQT.

2.5 Assumptions and Constraints

There are numerous assumptions and constraints that affect the development and operation of MAGIC. Limitations of both a probable and possible nature are discussed in the following subparagraphs.

2.5.1 Assumptions. The following assumptions are made concerning the design and development of MAGIC. They are categorized into five groups: user, system, maintenance, development, and hardware.

2.5.1.1 User Assumptions. These assumptions are related to what the individual user may expect from the design and development of MAGIC:

- a. The current GIPSY system will remain supported and operational until the H6000 is no longer used.
- b. The current GIPSYmate system will remain supported and operational until the WIS Workstation is no longer used.
- c. All current GIPSY functionality will continue to be available in MAGIC.
- d. The accuracy and validity of GIPSY processing will not be negatively impacted by the introduction of interfaces to MAGIC.
- e. MAGIC will be an interactive system.
- f. MAGIC will be an unclassified system which assumes user responsibility for file security classifications and/or data classifications.
- g. The current variety of graphic output devices, including numerous terminal types, and on-line printer, will continue to be available in MAGIC.
- h. The use of COTS packages to provide graphics generation in MAGIC will ease the transfer, editing, and manipulation of graphic pictures.

2.5.1.2 System Assumptions. These are assumptions about overall system considerations as they pertain to both MAGIC and the operating system on which it resides (Unix):

- a. GIPSY will remain supported and operational until MAGIC successfully completes all system integration tests and FQT.
- b. The functional baseline of GIPSY will not change during MAGIC's development.
- c. MAGIC will operate on a Unix-based workstation platform that will utilize POSIX-compliant interfaces to communicate with the H6000.

2.5.1.3 System Maintenance Assumptions. These assumptions pertain to specific considerations which are germane to the maintenance and support functions that must be performed by MAGIC system programmers:

- a. User support and system maintenance will be provided by Government and contractor personnel.
- b. Utilities used for system maintenance will have to be written for use in MAGIC.

2.5.1.4 System Developmental Assumptions. These assumptions summarize the underlying expectations that system programmers/designers may expect with regard to MAGIC's development effort:

- a. MAGIC will provide functional compatibility with all currently supported GIPSY applications.
- b. MAGIC will be built around logical CSCIs based on functionality.
- c. MAGIC will be implemented using a HOL that enforces structured programming and encourages data abstraction, information hiding, and meaningful names.
- d. The HOL used to develop MAGIC will provide easy readability.
- e. MAGIC development will be done using a combination of both DoD-STD-7935A and DoD-STD-2167A for software design and documentation. Development will be accomplished by means of the rapid application development paradigm.
- f. Software configuration management (SCM) of the effort will be based on policies and procedures discussed in SDP 2-90 (reference (n) of paragraph 1.2).
- g. MAGIC will be developed in an ASCII environment.
- h. MAGIC will integrate a number of COTS packages to provide the overwhelming bulk of system functionality.
- i. MAGIC will utilize X Windows and OSF/Motif to provide a user-friendly GUI.

2.5.1.5 Hardware Assumptions. These assumptions are related to considerations about the hardware platform upon which MAGIC will be implemented and fielded:

- a. MAGIC will be developed and fielded on a Unix-based workstation platform.
- b. Interfaces between the workstation and the WWMCCS host will be POSIX-compliant.

2.5.2 Constraints. There are several system dependencies and functional processing characteristics intrinsic to current GIPSY. However, since MAGIC is being implemented as an entirely new system that will only interface to GIPSY, these characteristics are not expected to constrain MAGIC's design.

It is expected that some constraints on MAGIC's design will arise from the planned integration of COTS packages to perform a significant portion of the functional processing. Those COTS packages that do not possess a programmer-level interface will be less integrated into MAGIC's overall system than those that do. Furthermore, when actually in a COTS package, the MAGIC user will be constrained to the services and functionality that the package provides.

SECTION 3. DETAILED CHARACTERISTICS

This section provides a detailed description of the functions to be performed by MAGIC and the attendant performance requirements.

3.1 Specific Performance Requirements

This paragraph contains a discussion of the various performance requirements to be used in the development of MAGIC. Functional requirements are grouped in appendix A sectioned by their appropriate CSCI.

3.1.1 Accuracy and Validity. In order to fulfill its primary mission of WWMCCS user support, MAGIC must provide processing services which are both accurate and valid. While accuracy refers to MAGIC's capability to provide display and processing functionality that is free from error, validity is associated with concepts or data that is founded on truth and/or fact.

3.1.1.1 System Accuracy. There are four specific areas of concern when dealing with system accuracy: correct user command syntax, data type checking, map data accuracy, and computational accuracy.

3.1.1.1.1 Correct User Command Processing. Due to the menu-driven, user-friendly concept central to MAGIC's operations, accurate processing of user-input commands/information is paramount as an overall system requirement. Much of this requirement will be satisfied by the routines utilized in the X Windows and Motif libraries to process menu selections, file selection boxes, and forms.

When the GUI is not being used (e.g., unassisted mode), some programmed routines must be used to process user input as well as some COTS package capabilities.

3.1.1.1.2 Data Type Checking. MAGIC checks all field names specified in user applications for compatibility of use within the constraints of that particular data type. That is, any data field used is checked for validity (based on the data type specified for that particular column) of use throughout the MAGIC session. Data types supported will be in compliance with the SQL-based Oracle COTS package.

3.1.1.1.3 Map Data Accuracy. The accuracy of map data is dependent upon the DeLorme Mapping System COTS package.

3.1.1.1.4 Computational Accuracy. MAGIC's computational accuracy will be determined primarily by the accuracy of the COTS packages being utilized to provide the specific functionality (e.g., Oracle, Wingz, DeLorme). Wherever MAGIC code is used for computations, however, they must be accurate to the limitations of the programming language and the host platform.

3.1.1.2 System Validity. There are two specific areas of concern when dealing with system validity: user command language syntaxing and user data considerations.

3.1.1.2.1 Valid User Command Processing. The capability to correctly process user-input commands in a valid manner will be handled primarily by the X Windows and Motif-based GUI. When the GUI has been bypassed, validity requirements will be satisfied by programmer-developed routines and the COTS packages' own command processing capabilities.

3.1.1.2.2 User Data Considerations. The issue of data validity is completely within the user's scope since MAGIC has no control over the data input by the user. The validity of any specific report, graphic or non-graphic, is entirely dependent upon the user's data, the system (workstation) environment, and the methodology employed to generate the report.

3.1.2 Timing. Timing requirements for MAGIC are twofold:

- a. MAGIC's response to a user's mouse click or a keystroke for a menu or dialog box shall be within a 5 second timeframe.
- b. If the user-input choice requires MAGIC to interface with a COTS package (either launching or processing), system response shall be within a 5 second timeframe. In other words, the users must either receive some sort of acknowledgment that processing is going on or obtain the end result of their selection.

3.1.3 Capacity Limits. Sizing requirements pertinent to MAGIC are:

- a. A minimum of 8 megabytes (Mb) of Random Access Memory (RAM) is required to execute MAGIC (more RAM would permit faster execution).
- b. A minimum of 2 Mb of free disk space is be required to execute MAGIC.
- c. A minimum of 32 Mb of swap space is required to execute MAGIC.

3.2 Functional Area System Functions

This paragraph performs a dual purpose: (1) it provides additional details about the individual functions of the major functional processing steps specified in paragraph 2.4 and (2) it relates the functions to the performance requirements discussed in paragraph 3.1.

3.2.1 Graphical User Interface. MAGIC will utilize a graphical user interface (GUI) to provides the functional ability to interact with the user by means of a Motif-based GUI. The major functional processing includes: (1) system initialization, (2) the creation of menus, screens, and dialog boxes that permit the user to make selections with either keystroke or mouse click, (3) the processing of user input to include validation and verification of

that input, and (4) calling other MAGIC CSCIs and COTS packages as needed to satisfy user requests for action.

This function will also be able to recognize that a user wishes to: (1) bypass normal GUI processing and work directly with a COTS package or (2) use xterm to communicate with the host platform.

The following performance requirements are relevant to this function:

- a. Accuracy and validity
 - (1) System accuracy
 - (a) Correct user command processing
 - (b) Data type checking.
 - (2) System validity
 - (a) Valid user command processing
 - (b) User data considerations.
- b. Timing
- c. Capacity limits.

3.2.1.1 Context-Sensitive Help. This function will be used to provide a user-friendly, context-sensitive help facility to the user. Help may be requested through either keystroke or mouse click from the standard Motif-based GUI screens unless superseded by a COTS package or xterm processing.

The following performance requirements are relevant to this function:

- a. System accuracy with regard to correct user command processing
- b. System validity with regard to valid user command processing
- c. Timing
- d. Capacity limits.

3.2.1.2 Error Detection/Handling. This function will be used to provide interactive error handling and correction when detected through normal processing. Error handling is provided with the standard Motif-based GUI screens unless superseded by a COTS package or xterm processing.

The following performance requirements are relevant to this function:

- a. System accuracy

- (1) Correct user command processing
- (2) Data type checking.

b. System validity with regard to valid user command processing.

3.2.2 Data Retrieval. MAGIC will enable the user to access data from user databases located on either the WWMCCS host or the Unix workstation. The workstation databases may be Oracle relational database tables to which the user has access or flat ASCII files. Data in the databases is selected by a user according to a qualification criterion; the resulting data subset can then be manipulated and presented as user-formatted reports, statistical graphs, and geographic displays.

The following performance requirements are relevant to this function:

a. Accuracy and validity

(1) System accuracy

(a) Data type checking

(b) Computational accuracy

(2) System validity with regard to user data considerations.

b. Timing

c. Capacity limits.

3.2.3 Business Reports. MAGIC will enable the user to create and display reports. Each report will consist of data selected from a previously identified MAGIC internal format subset of a database. Thus, the user will have the flexibility to display a report in a form most suited to his/her specific needs, varying from simple formatted reports to line graphs and pie charts. The Wingz COTS package will be used to provide two levels of support: an assisted mode, which helps the user in constructing reports and graphs, and an unassisted mode in which the user is simply placed within the Wingz product and is expected to know what is needed to use that package effectively.

The following performance requirements are relevant to this function:

a. Accuracy and validity

(1) System accuracy with regard to computational accuracy

(2) System validity with regard to user data considerations.

b. Timing

- c. Capacity limits.

3.2.4 Geographic Displays. MAGIC will provide the user with an integrated mapping capability that utilizes a DeLorme Mapping System (DMS) possessing the following capabilities:

- a. All digital mapping system with data resolution to 1 centimeter
- b. Capable of displaying vector data, scanned paper map data, scanned aerial photography data, digital satellite imagery, either separately or merged
- c. Capable of accessing and displaying digital geographic data at any location and level of detail in an average of 6 seconds or less from a database of many gigabytes
- d. Capable of linking with an external relational database
- e. Global vector database at 1:3,000,000 level of detail, with select areas at higher detail
- f. Capable of centering the screen at any location on the Earth's surface
- g. Full declutter capabilities for vector data
- h. Display data in 22 levels, from a world view to a view that shows an area 40 feet across
- i. Overlay capabilities with text, lines, fills, circles, boxes, and military, cartographic, and user-defined symbols
- j. Overlays are geographically referenced for proper placement under pan and zoom operations
- k. Special utilities such as Great Circle route computation, latitude/longitude and Universal Transverse Mercator (UTM) grids, projections, etc.

The following performance requirements are relevant to this function:

- a. Accuracy and validity
 - (1) System accuracy
 - (a) Data type checking
 - (b) Map data accuracy
 - (c) Computational accuracy.

(2) System validity with regard to user data considerations.

b. Timing

c. Capacity limits.

3.2.5 Graphic Editing. MAGIC will provide the user with freehand drawing functions as well as the interactive capability to create new slides, enhance existing slides, and edit multiple slides simultaneously.

The following performance requirements are relevant to this function:

a. Accuracy and validity

(1) System accuracy

(a) Correct user command processing

(b) Computational accuracy.

(2) System validity with regard to valid user command processing.

b. Timing

c. Capacity limits.

3.2.6 Slide Presentations. Through this function, MAGIC will provide the user with access to two types of management capabilities: (1) maintain an inventory of slides and (2) the ability to organize slides into a briefing.

The following performance requirements are relevant to this function:

a. Accuracy and validity

(1) System accuracy

(a) Correct user command processing

(b) Computational accuracy.

(2) System validity with regard to valid user command processing.

b. Timing

c. Capacity limits.

3.2.7 Internal System Functions. In addition to the above explicit functions performed by MAGIC, there are also capabilities that are hardware-dependent or required by more than one CSCI. This function provides a multitude of services unseen by the user but critical to MAGIC functionality. As its name

implies, a variety of low-level services that tie the various functional CSCIs together are provided. They are as follows:

- a. MAGIC environment initialization and cleanup
- b. File input/output (I/O) functions such as open, close, read, and write operations
- c. Functions useful to a number of MAGIC's CSCIs, such as string utilities and pathname processing
- d. Communications between the Unix workstation and the WWMCCS host platform.

The following performance requirements are relevant to this function:

- a. Accuracy and validity
 - (1) System accuracy
 - (a) Correct user command processing
 - (b) Data type checking.
 - (2) System validity
 - (a) Valid user command processing
 - (b) User data considerations.
- b. Timing
- c. Capacity limits.

3.2.8 System Programmer Functions. This function is targeted to the MAGIC maintenance programmer to perform system development and maintenance functions. The following capabilities will be performed:

- a. Provide statistical reports about MAGIC usage
- b. Generate printed listings of MAGIC source code
- c. Identify differences between two MAGIC source code files
- d. Update MAGIC user help and error messages
- e. Assist in the compilation/linking of MAGIC's code when generating executable files (e.g., makefile).

- f. Supported the automated generation of MAGIC documentation based on both source code structures, comments, and header comment blocks.

The performance requirements identified in paragraph 3.1 are not applicable to this function.

3.3 Inputs and Outputs

This paragraph normally would describe each data input and output projected for usage in MAGIC. However, due to the overwhelming preponderance of COTS packages in what will become MAGIC and the corresponding lack of knowledge about the number, type, and range of elements used by each package, this paragraph has been tailored out.

3.4 Database/Data Bank Characteristics

This paragraph discusses the data elements to be used in MAGIC's database operations. Although MAGIC has no database of its own, database operations may be initiated, performed, and/or coordinated by MAGIC on databases from the three external sources described below.

3.4.1 Host-Based User Data. MAGIC will be able to access a user's data file that resides on the WWMCCS host through an interface to GIPSY. Data elements specific to this database source are listed below. All have been previously described in subparagraph 2.3.2.10:

- a. FDT
- b. GDS
- c. QDF
- d. QDT
- e. Index File (User Index File).

3.4.2 Workstation-Based Databases. There are two projected databases for the workstation platform. MAGIC will utilize Oracle for user-specific database operations and DeLorme for mapping database operations.

3.4.2.1 Oracle Relational Database Management System (RDBMS). MAGIC will be able to access a user's data file that resides on the workstation as either an ASCII file or one stored in Oracle COTS format. Data manipulation will be accomplished using Oracle which also includes data downloaded from the WWMCCS host for local use. Data elements specific to this database source are listed below:

- a. User Data File - either a Unix-based ASCII file residing in some directory on the workstation or Oracle-based table(s) that reside within the static Oracle COTS partition.

- b. Data Description - a file that describes the number of columns in a data file to include column width and data type. The supported data types will be appropriate to Oracle's implementation of SQL.
- c. Data Link File - a file that captures the possible relationships between the data descriptions and data files in MAGIC (e.g., a data file can be described in more than one way).

3.4.2.2 DeLorme Mapping System (DMS). MAGIC will be able to access a digital mapping database available as the DeLorme COTS package. The map database will reside on the workstation and be capable of accepting periodic updates and permitting overlay of retrieved user data on map displays. Data elements specific to this database source are listed below:

- a. Map File - the file within the DeLorme product containing the digitized data points required to generate geographic and geodetic map displays on the workstation.
- b. Overlay Data - the file containing the set or subset of retrieved user data to be overlayed on the DeLorme-generated map display.
- c. Map Link File - a file that captures the linkage between a previously stored map and the overlay data that is stored separately. Usage of the map link file permits the subsequent regeneration of the previously generated (and displayed) geographic display.

3.5 Failure Contingencies

Regular and complete MAGIC development system backups will be made to magnetic tape (Unix tar format) on a weekly basis. This ensures that any MAGIC system files that may be accidentally deleted, overwritten, damaged, or improperly changed may be recovered. Any changes made to a file between the time of the backup and the time it is lost will, of course, not be recovered and would have to be reconstructed. For the most part, however, all system files may be restored if necessary.

At fielded sites, system backups become the responsibility of the system administrator for the workstation(s) in question. MAGIC's user documentation will warn users and urge them to perform regular system backups or see that an individual in authority does for them.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 4. DESIGN CONSIDERATIONS

This section describes how MAGIC's development will satisfy the requirements delineated in sections 2 and 3. The user's requirements for MAGIC are included here and stated in technical terminology.

4.1 System Description

MAGIC has evolved from the modernization of the WWMCCS standard host-based GIPSY and the Z-248 PC-based modernized GIPSYmate system. Designed and developed to meet the needs of a new generation of WWMCCS users, MAGIC is to be fielded as a resident system on the Macintosh IIcx workstation and will present the user with a menu-based graphical user interface (GUI) that integrates (as transparently as feasible) the commercial off-the-shelf (COTS) packages also resident on that platform. Processing facilities appropriate to both sophisticated and novice users will be supported as well as the ability to access the full range of data base types found on the WWMCCS host. Functionally, the MAGIC user will have the capability to perform data retrieval and manipulation operations, business graphics displays, geographic and geodetic mapping displays, slide show generation, and graphic editing operations.

Specifically, MAGIC will utilize the workstation-resident Oracle COTS package for its DBMS-related operations and data files on both the workstation and an H6000 host may be accessed. Furthermore, MAGIC operations related to data management may be accomplished through any one of three methods: (1) Assisted Structured Query Language (SQL) which is the default menu-driven method, (2) Unassisted SQL which bypasses the menus and permits the free-form entry of sophisticated SQL queries, and (3) Process Control Statement (PCS) Mode which also bypasses the menus but accepts GIPSY language PCS statements for uploading to the H6000 host for remote processing.

MAGIC's business graphics capability will be provided through the Wingz product available from Informix, Inc. MAGIC's geographic and geodetic mapping functions will be provided through an additional COTS package--the DeLorme Mapping System. These and other COTS packages resident on the workstation will be integrated through MAGIC's GUI.

4.2 System Functions

The following subparagraphs describe MAGIC's functions in sufficient detail that the discussion in section 5 can related to them. Specifically discussed are the performance requirements (first stated in paragraph 3.1) and the system's functions (first described in paragraph 3.2).

4.2.1 Performance Requirements. The following subparagraphs describe MAGIC's requirements for accuracy and validity, timing, and capacity.

4.2.1.1 Accuracy and Validity. MAGIC has the following performance requirements as they related to accuracy and validity:

- a. The system must be correct. That is, MAGIC is expected to satisfy its specifications and fulfill the user's mission objectives.
- b. The system must be reliable. MAGIC is expected to perform its intended functions with error tolerance of no greater than 2 percent, consistency, accuracy, and simplicity.
- c. The system must have integrity. MAGIC must be access controlled and access auditable.
- d. The system must be usable, maintainable, testable, portable, reusable, interoperable, and traceable.

4.2.1.2 Timing. MAGIC has the following timing requirements:

- a. The system's response to a user's mouse click or a keystroke for a menu or dialog box shall be within a 5 second timeframe.
- b. If the user-input choice requires MAGIC to interface with a COTS package (either launching or processing), system response shall be within a 5 second timeframe. In other words, the user must either receive some sort of acknowledgment that processing is going on or obtain the end result of their selection.

4.2.1.3 Capacity Limits. MAGIC has no specific maximum capacity limitations beyond how utilization of capacity may adversely affect the performance requirements delineated in subparagraph 4.2.1.2. Minimum capacity limits are noted below:

- a. A minimum of 8 Mb of RAM
- b. A minimum of 2 Mb of free disk space
- c. A minimum of 32 Mb of swap space.

4.2.2 Functional Requirements. The following subparagraphs elaborate on the system functions described in paragraph 3.2. In order to provide a more exact picture of how MAGIC is to be designed and developed, the subparagraphs will be geared toward specific CSCI (subsystem) names.

4.2.2.1 Human Interface. The Human Interface CSCI will function as the logical hub of all MAGIC processing activities and present a user-friendly graphical user interface (GUI) to the user that is compliant with Open Software Foundation (OSF)/Motif. The user may choose menu selections with either keystrokes or mouse clicks. Dialog boxes are provided for those instances when the user is required to provide additional information before MAGIC can continue processing.

When a user initiates a MAGIC session, the program menus will be presented to the user, and control of program actions will begin in this CSCI. As users navigate through the menu system to an activity they want to perform, control is eventually passed to the appropriate functional CSCI (COTS package), such as Data Management or Business Graphics, to perform the desired activity. The Human Interface CSCI will also provide context-sensitive help that is compliant with Motif applications.

4.2.2.2 Data Management. The Data Management CSCI will enable the user to access data from user databases located on either the WWMCCS host or the workstation (Sun or Macintosh). The workstation databases may be Oracle relational database tables to which the user has access or ASCII files. Host-based user data can be any file type accessible by the host-based GIPSY application.

Data in the databases is selected by a user according to qualification criteria; the resulting data subset can then be manipulated and presented as user-formatted reports, statistical graphs, and geographic displays. MAGIC will emulate common SQL operations and format in the assisted mode and will support the full range of Oracle and/or GIPSY functionality in unassisted mode.

4.2.2.3 Business Graphics. The Business Graphics CSCI will enable the MAGIC user to create and display reports. Each report may consist of data selected from a previously-identified MAGIC retrieval (assuming one was accomplished) or the user will be able to enter raw data directly into the spreadsheet format used by the COTS product.

This CSCI will provide the user with the flexibility to display a report in a form most suited to the user's individual needs, varying from simple formatted reports to line graphs and pie charts. The Wingz COTS package will be used to provide the user with two levels of support: an assisted mode, which helps the user in constructing reports and graphs, and an unassisted mode in which the user has access to the full set of Wingz' capabilities (based on the assumption that the MAGIC user is able to use the Wingz product effectively).

4.2.2.4 Geographic Mapping. The Geographic Mapping CSCI will enable the MAGIC user to create, edit, and display both vector- and raster-based maps with an optional ability to overlay the user's data on the workstation. The DeLorme Mapping System (DMS) will be used to provide the map data and the overwhelming bulk of functionality for this CSCI with MAGIC providing a user-friendly graphical user interface (GUI) to the underlying COTS package.

Map data will be retrieved and displayed from large digital databases. Once retrieved, any location in the world may be specified and all geographic data in the system may be located precisely in terms of its latitude and longitude or Universal Transverse Mercator (UTM) position.

Since the CSCI will display geographic information based on its level of importance, the user can zoom down from a whole world view with only

continents and oceans displayed to very small areas with street names and buildings shown (where the applicable data is available).

4.2.2.5 Graphic Editor. The Graphic Editor CSCI will provide an interactive capability to create new slides, enhance existing slides, and edit multiple slides simultaneously on the workstation through the use of freehand drawing functions, polygon manipulation, and other graphic editing techniques. The resultant slides may be viewed singly by the user in a MAGIC session or incorporated into a slide presentation with other slides.

The use of a COTS package for this CSCI is undecided at this time. Based on appropriate information at the time this CSCI is designed and developed, the Graphic Editor may be wholly-developed code or integrated as a COTS package.

4.2.2.6 Slide Show. The Slide Show CSCI will provide access to two types of management capabilities. First, it will allow the user to maintain an inventory of slides; and second, it will provide the user with the ability to organize slides into a briefing on the workstation.

Additionally, this CSCI will provide import and export facilities for transfer of slides and briefings between MAGIC and compatible COTS packages. The Slide Show CSCI will also support a hardcopy capability for slides and briefings.

The use of a COTS package for this CSCI is undecided at this time. Based on appropriate information at the time this CSCI is designed and developed, the Slide Show may be wholly-developed code or integrated as a COTS package.

4.2.2.7 Internal Processing. The Internal Processing CSCI will provide capabilities that are hardware-dependent or required by more than one CSCI. As such, this CSCI will serve as the system toolbox, the common ground for general routines used elsewhere, and the repository of less portable routines. Whenever possible, this CSCI will be used to provide an interface layer between MAGIC and the outside world unless such a layer adversely affects system performance for very little engineering gain.

4.2.2.8 Programmer Utilities. The Programmer Utilities CSCI will provide a number of tools and functions designed to aid the MAGIC maintenance programmer. The functions of this CSCI will provide automated documentation support, support the gathering of metrics, assist in system release activities, and support low-level programmer operations (e.g., comparing source files).

4.3 Flexibility

The flexibility requirements for MAGIC are:

- a. The system must be modular. As such, MAGIC should be highly cohesive, very loosely coupled, designed with low complexity, and utilize structured programs.

- b. MAGIC must be general--it should not have input, processing, and output functions mixed in the same modules. All constants should be defined only once in the system. Application and machine-dependent functions should not be mixed in the same modules.
- c. MAGIC must be expandable. CSCIs must perform logical processing independent of data storage specifications (not commit all available memory capacity) and be extensible in terms of computational functions.
- d. MAGIC's source code must be self-descriptive with sufficient comments to explain the implementation of a function.

4.4 System Data

As previously noted in paragraph 3.3, input and output data was tailored out of this document due to the overwhelming preponderance of COTS packages in MAGIC and the corresponding lack of knowledge about the number, type, and range of elements used by each package.

As previously noted in subparagraph 3.4.1, detailed information regarding the five host-based user data elements were discussed previously in subparagraph 2.3.2.10.

Of the workstation-based data elements described in subparagraph 3.4.2, the default structure and encoding is projected to be Unix ASCII strings. Specific deviations from this default will be required to address the requirements of particular databases (e.g., Oracle).

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 5. ENVIRONMENT

This section describes the current ADP environment and projects the environment needed to satisfy the requirements delineated in sections 2 and 3.

5.1 Equipment Environment

This paragraph describes the equipment capabilities required for the operation of the proposed system--MAGIC. General descriptions of currently available equipment as well as characteristics of the new equipment necessary (based on the information in section 3) will be presented.

5.1.1 Current Hardware Environment. GIPSY resides on a Honeywell H6000/DPS8000 mainframe. It operates under the Honeywell GCOS 8 operating system, which provides both interactive and batch processing capabilities in a WWMCCS (TEMPEST-protected) environment. Either magnetic tape or magnetic disk space on the H6000 may be utilized for storage purposes.

A minimum hardware configuration for GIPSY consists of an H6000 with sufficient memory to execute a 35,000 word (approximately 205 Kb) program; work and permanent file space; and an alphanumeric I/O terminal. The optimal configuration includes a graphics I/O terminal and a program-controlled hardcopy device. All GIPSY functions except actual graphics output can be performed on the minimum configuration.

GIPSY currently supports a variety of graphic output devices including numerous terminal types and an on-line printer.

5.1.1.1 Supported Terminal Types. GIPSY currently supports 20 different terminal types. They are identified and described to GIPSY via the System Definition (SYSDEF) File. The supported terminal types are:

- a. Tektronix 4014-1
- b. VIP 7705
- c. VIP 786W
- d. IBM 2741
- e. KSR 33 Teletype
- f. Tektronix 4012-1
- g. Tektronix 4051
- h. Tektronix 4014-1 with intelligence
- i. Tektronix 4027

- j. Batch GIPSY
- k. Tektronix 4014-1 over WIN
- l. UNIVAC 1652 with graphics
- m. Tektronix 4107
- n. Standard WSGT
- o. Tektronix 4054
- p. WSGT (Synchronous)
- q. WSGT (Phase III)
- r. WIS Workstation (IBM 3270 PC/XT)
- s. Zenith Z-248 PC/AT
- t. Sun SPARCstation (in Tektronix 4014 emulation mode).

5.1.1.2 On-Line Printer. Non-graphics output may be directed to any on-line printer that has an ASCII configuration.

5.1.2 Proposed Hardware Environment. The MAGIC development effort projects two different hardware environments--prototype and target. The following subparagraphs present those two configurations.

5.1.2.1 Prototype Environment. This subparagraph identifies those hardware and firmware items that will comprise MAGIC's prototype environment on the Sun SPARCstation. There are no security issues associated with these items. The following list describes the purpose of each item:

- a. Hayes-compatible modem (2400 baud) - the highest speed modem type that is compatible with the Defender access software used to obtain H6000 access.
- b. Honeywell 6080 (unclassified host) - the WWMCCS-like mainframe host platform used to support MAGIC execution in an unclassified environment.
- c. PostScript printer (optional) - the optional printer that may be used to obtain hardcopies of graphic displays generated by the Wingz COTS package.
- d. Sun Desktop Backup Pack, DC6150 media - the tape backup unit used to install MAGIC software from the Unix tar-formatted tape.

- e. Sun cg3/6 color monitor - the color monitor used to support color graphics on the Sun workstation.
- f. Sun SPARCstation - the workstation platform used to execute MAGIC software in an unclassified environment.
- g. 2 megabytes (Mb) free disk space - the amount of disk space needed to permit the installation of MAGIC software.
- h. 8 Mb of random access memory (RAM) - the amount of available RAM needed to execute MAGIC on the Sun workstation.
- i. 32 Mb swap space - the amount of swap space needed to permit efficient execution of MAGIC software.

5.1.2.2 Target Environment. This subparagraph identifies those hardware and firmware items that will comprise MAGIC's target environment on the Macintosh IIfx workstation. The following list describes the purpose of each item and notes any pertinent security issues:

- a. Apple Macintosh IIfx - the workstation platform used to execute MAGIC software in a classified WWMCCS environment.
- b. Color monitor - the color monitor used to support color graphics on the Macintosh workstation.
- c. Honeywell 6080/DPS8000 (WWMCCS host) - the WWMCCS mainframe host platform used to support MAGIC execution in a classified environment.
- d. Postscript printer (optional) - the optional printer that may be used to obtain hardcopies of graphic displays generated by the Wingz COTS package.
- e. Tape backup unit - the tape backup unit used to install MAGIC software from the Unix tar-formatted tape.
- f. 2 Mb free disk space - the amount of disk space needed to permit the installation of MAGIC software.
- g. 8 Mb of RAM - the amount of available RAM needed to execute MAGIC on the Macintosh workstation.
- h. 32 Mb swap space - the amount of swap space needed to permit efficient execution of MAGIC software.

5.2 Support Software Environment

This paragraph describes the support software with which MAGIC will interact. As noted previously, MAGIC development will occur in two phases: prototype and

target. As such, there are two projected support software environments that are presented in the following subparagraphs.

5.2.1 Prototype Environment. This subparagraph identifies those software support items that will comprise MAGIC's prototype environment on the Sun SPARCstation. There are no security issues associated with these items. The following list describes the purpose of each item:

- a. Defender access software - the software known as "Defender" that screens and enforces access control via modem connection to a number of ADP platforms, including the unclassified H6000 at the Operational Support Facility (OSF) in Sterling, Virginia.
- b. DeLorme Mapping System (XMAP Release 1.1) - the COTS package required for use in providing interactive mapping capabilities to the MAGIC user.
- c. Graphic Information Presentation System (GIPSY), Release 5.2 - the host-based software system that provides MAGIC users with access to their data that resides in host databases.
- d. Honeywell Datanet software - the software that supports and controls communications through the Honeywell Datanet.
- e. Honeywell GCOS 8 (unclassified) - the Honeywell operating system needed to provide a WWMCCS-like environment and access to host functionality in an unclassified environment.
- f. Honeywell Time Sharing System (TSS) - the Honeywell subsystem that provides host-based access to all other subsystems and services on the H6000 platform (including the ability to initiate GIPSY).
- g. Informix Wingz, Release 1.0 (for Sun) - the COTS package required for use in providing interactive business-related graphics capabilities to the MAGIC user.
- h. MAGIC, Release 1.0 - the workstation-based application software that provides access to the host-based GIPSY and integrates the workstation-based COTS packages with a Motif-based GUI.
- i. Open Software Foundation (OSF)/Motif Window Manager, Release 1.0.A - the window manager for the required GUI look-and-feel used in MAGIC. Also, the release compatible with the X Window Server noted in (o) below.
- j. Oracle Relational Database Management System (RDBMS), Version 6.0.27 - the COTS package required for use in providing data retrieval, manipulation, and report capabilities to the MAGIC user.

- k. Oracle SQL*Loader, Version 1.0.18 - the COTS package used in conjunction with the Oracle RDBMS to permit Oracle's import of data from outside sources.
- l. Oracle SQL*Plus, Version 3.0.6.5.1 - the COTS package used in conjunction with the Oracle RDBMS to support Oracle's implementation of Standard SQL with some extensions.
- m. SunOStm, Release 4.0.3c - the Unix operating system on the Sun platform needed to provide the Unix environment that MAGIC requires.
- n. Sun OpenWindows, Version 1.0 - the Sun proprietary COTS windowing package that the Sun version of Wingz requires in order to execute.
- o. X Window Server, Version 11, Release 3 - the window server required by the Motif window manager and handler for various Xlib function calls used in support of the DeLorme package and specific low-level processes within MAGIC.

5.2.2 Target Environment. This subparagraph identifies those software support items that will comprise MAGIC's target environment on the Macintosh IIx workstation. The following list describes the purpose of each item and notes any pertinent security issues:

- a. Communication Transport Layer Interface (CTLI) Software - the Macintosh-resident secure software that permits and supports communications between that platform and the WWMCCS host.
- b. DeLorme Mapping System - the COTS package required for use in providing interactive mapping capabilities to the MAGIC user.
- c. GIPSY Release 5.2 - the host-based software system that provides MAGIC users with access to their data that resides in host databases.
- d. Honeywell Datanet software - the software that supports and controls communications through the Honeywell Datanet.
- e. Honeywell GCOS 8 (WWMCCS version) - the Honeywell operating system needed to provide a WWMCCS environment and access to host functionality in a classified environment.
- f. Honeywell TSS - the Honeywell subsystem that provides host-based access to all other subsystems and services on the H6000 platform (including the ability to initiate GIPSY).
- g. Informix Wingz, Release 1.0 (for Macintosh) - the COTS package required for use in providing interactive business-related graphics capabilities to the MAGIC user.

- h. Macintosh A/UX 2.0 - the secure Unix operating system on the Macintosh platform needed to provide the Unix environment that MAGIC requires to execute in a WWMCCS environment.
- i. MAGIC, Release 1.0 - the workstation-based application software that provides access to the host-based GIPSY and integrates the workstation-based COTS packages with a Motif-based GUI.
- j. OSF/Motif Window Manager - the secure window manager (on the Macintosh) for the required GUI look-and-feel used in MAGIC. Also, the release compatible with the X Window Server noted in (n) below.
- k. Oracle RDBMS - the COTS package required for use in providing data retrieval, manipulation, and report capabilities to the MAGIC user.
- l. Oracle SQL*Loader - the COTS package used in conjunction with the Oracle RDBMS to permit Oracle's import of data from outside sources.
- m. Oracle SQL*Plus - the COTS package used in conjunction with the Oracle RDBMS to support Oracle's implementation of Standard SQL with some extensions.
- n. X Window Server - the secure window server (on the Macintosh) required by the Motif window manager and handler for various Xlib function calls used in support of the DeLorme package and specific low-level processes within MAGIC.

5.3 Communications Requirements

The following subparagraphs discuss the projected communications requirements for the MAGIC development effort.

5.3.1 Graphic Overview. The known communications requirements of the MAGIC system appear graphically as figures 5-1 and 5-2.

5.3.2 Hardware. The communications hardware known to be required in support of MAGIC differs between the two development phases. While a prototype system, MAGIC will utilize a remote access connection to an unclassified H6000/DPS8000 platform. As a consequence, the following hardware will be needed:

- a. 4 - Hayes-compatible modems (2400 baud)
- b. 4 - dedicated phone lines
- c. 1 - dial-in H6000 port (minimum).

Upon transition to the target environment, MAGIC will utilize a direct connection to the WWMCCS host platform--a classified H6000/DPS8000. To provide communications services for this configuration, the following hardware

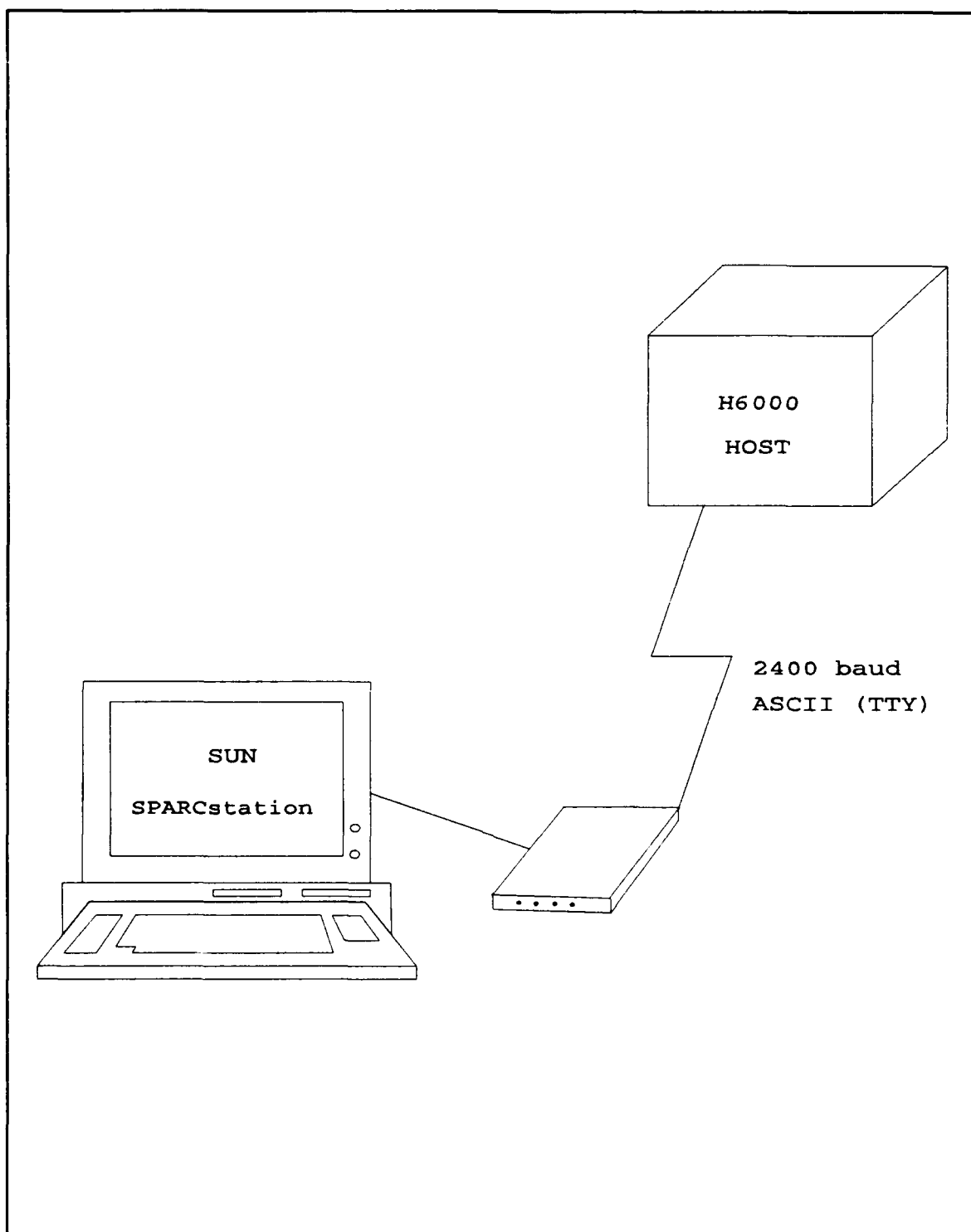


Figure 5-1. Prototype Communications Requirements

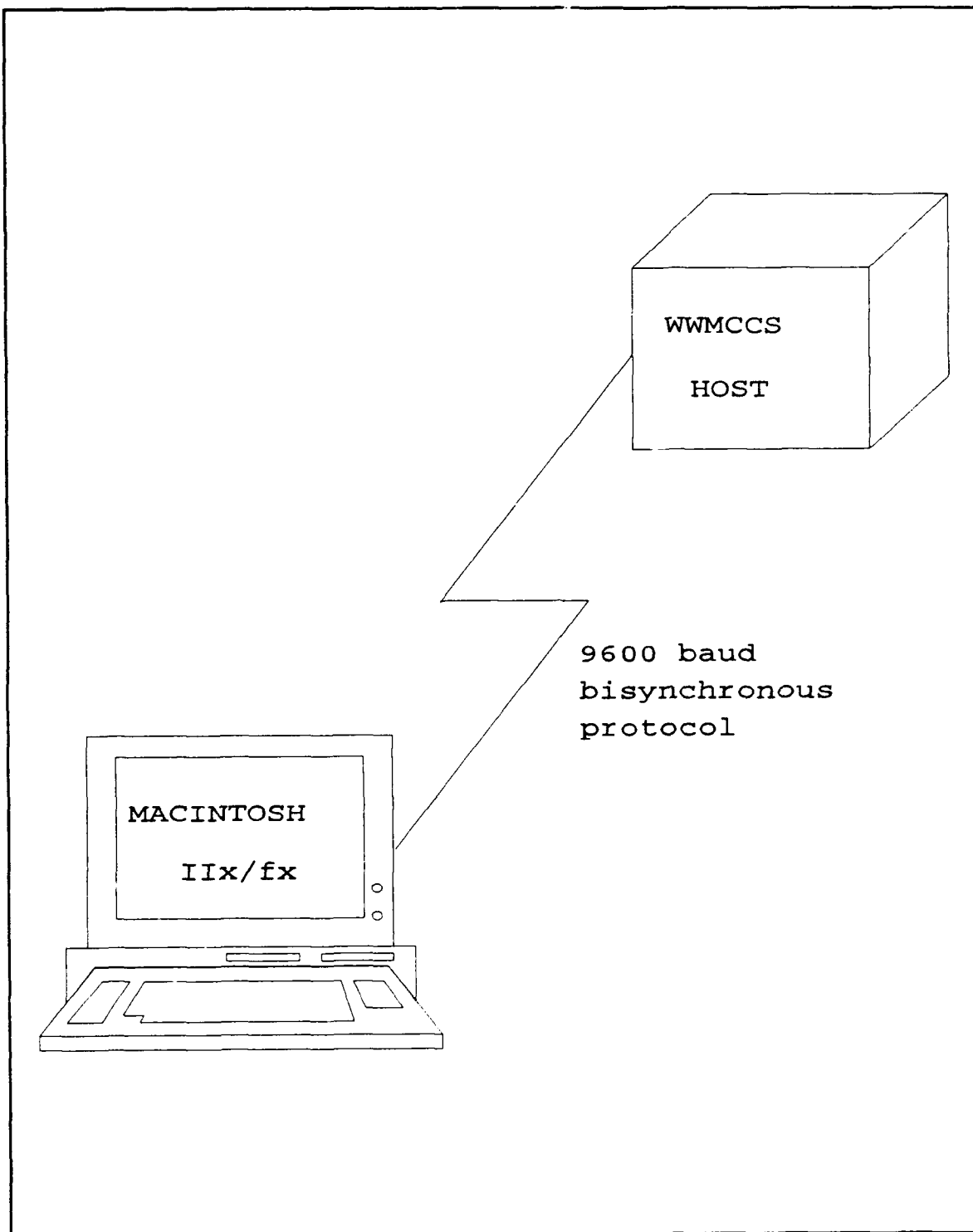


Figure 5-2. Target Communications Requirements

will be needed by all WWMCCS sites with the same hardware required for each workstation to be used for MAGIC system operations:

- a. 1 - H6000 Datanet communications line
- b. 1 - direct connect port (switchable as an option).

5.3.3 Software. The communications software known to be required in support of MAGIC differs between the two development phases. As noted previously, MAGIC will utilize a remote access connection to an unclassified H6000/DPS8000 platform as a prototype. Thus, the following software will be needed:

- a. Modem-specific C code for TTY-based communications
- b. C code to monitor line time-out and to ensure data validity (e.g., Cyclical Redundancy Check (CRC) algorithm)
- c. Defender-specific C code needed to interact with the Defender access software on the H6000
- d. C code needed to perform logon and logoff services as well as the Motif-based screens to support such operations
- e. X server (event handler) and Motif Window Manager.

Upon transition to the target environment, MAGIC will utilize a direct connection to the WWMCCS host platform--a classified H6000/DPS8000. To provide communications services for this configuration, the following software will be needed:

- a. H6000 Datanet server software
- b. C code to monitor line time-out. The previous configuration's C code used to ensure data validity is not critical since the bisynchronous protocol used in direct connect configurations already includes similar checks
- c. C code needed to perform logon and logoff services as well as the Motif-based screens to support such operations
- d. Secure X server (event handler) and secure Motif Window Manager.

5.4 Interfaces

This paragraph describes all interfaces with other applications systems. Four known interfaces are described although additional interfaces are also projected to as-yet-undetermined COTS packages in support of the Graphic Editor and Slide Show CSCIs.

5.4.1 Oracle RDBMS. MAGIC's interface with Oracle (used as MAGIC's DBMS) is manual (the user must select data management-related operations and MAGIC will initiate the connection) and is used for two distinct purposes: data description and data manipulation. As such, MAGIC will interact both with the SQL*Loader (for the purpose of uploading data and data descriptions to Oracle for processing) and to Oracle itself (in order to manipulate and view the data).

Since both applications reside on the workstation, data transfer is dependent only on the clock speed of the workstation (in addition to the known effects of resident WWMCCS/WWMCCS Intercomputer Network (WIN)-approved security packages). A programmer's interface with MAGIC C source code is utilized to provide the interface and data transfer formats will comply with Oracle's requirements.

5.4.2 Wingz. MAGIC's interface with Wingz (spreadsheet software by Informix Software, Inc.) is manual (the user must select business graphics-related operations and MAGIC will initiate the connection). When selected, MAGIC will automatically reformat the current data set into Wingz-compliant format and load it into the COTS package for immediate user access.

Since both applications reside on the workstation, data transfer is dependent only on the clock speed of the workstation (in addition to the known effects of resident WWMCCS/WIN-approved security packages). A programmer's interface is not provided with Wingz' software so program launch will be achieved through usage of C function calls to the Unix kernel. Data transfer formats will comply with both Wingz' and the Unix kernel's requirements.

5.4.3 DeLorme Mapping Systems. MAGIC's interface with DeLorme is manual (the user must select geographic mapping-related operations and MAGIC will initiate the connection). When selected, MAGIC will ensure that the current data set is formatted in accordance with DeLorme's requirements and made available for immediate user access in the COTS package.

Since both applications reside on the workstation, data transfer is dependent only on the clock speed of the workstation (in addition to the known effects of resident WWMCCS/WIN-approved security packages). A programmer's interface with MAGIC C source code is utilized to provide the interface and data transfer formats will comply with DeLorme's requirements.

5.4.4 Graphic Information Presentation System (GIPSY). MAGIC's target interface with GIPSY (as noted in subparagraph 5.3.3 above) will utilize a direct connection to the Datanet of the WWMCCS host (H6000). The line speed supported is dependent upon that of the Datanet communications link and not MAGIC's software. The data transfer will be accomplished using a standard bisynchronous protocol and security requirements will be satisfied in accordance with accepted WWMCCS/WIN security procedures in effect at all WWMCCS ADP installations. Neither GIPSY nor MAGIC have any special security requirements since both are unclassified software systems that may be used to manipulate classified data.

The data transfer between the two systems will be accomplished by sending blocks of data with framing characters used by MAGIC to denote the type of data being received from GIPSY. The interface with GIPSY is strictly manual in nature. The MAGIC user must consciously decide to logon to the WWMCCS host for the purposes of using GIPSY processing (for host-based data retrieval operations).

Although initially implemented as a modem-based teletype (TTY) interface while MAGIC is in prototype development, this type of access is interim and is not planned to be supported when MAGIC is fielded.

5.5 Summary of Impacts

This paragraph describes anticipated organizational, operational, and development impacts of MAGIC's development on the WWMCCS community and DSSO.

5.5.1 ADP Organizational Impacts. There are no known organizational impacts related to MAGIC's development. Personnel responsibilities as well as numbers and skills of relevant personnel are discussed in the Software Development Plan (SDP 2-90).

5.5.2 ADP Operational Impacts. Operational procedures at the various ADP centers will be significantly affected by the fielding of MAGIC. Currently, there are very few Unix-based applications in the WWMCCS community and MAGIC is targeted for a distributed processing environment of either Sun SPARCstations or Macintosh IIx platforms. These workstations operate in stand-alone mode by default, may optionally connect to the host platform, and may be connected via a Local Area Network (LAN).

The WWMCCS community has just started redirecting itself toward distributed processing, Unix, and LANs and, as such, has very few defined operational guidelines and procedures in place. However, since the entire Joint community has been directed and is moving in that direction, the impacts felt will not be due solely to MAGIC's development and fielding. MAGIC's appearance on the scene may cause the impacts to be felt a little sooner (because MAGIC is in the vanguard of similarly planned products) than may have otherwise happened but they would have happened nonetheless.

5.5.3 ADP Development Impacts. The requirements for personnel and resources needed to develop and test MAGIC are addressed in subparagraphs 8.3.1 and 8.4.1 of this document.

5.6 Failure Contingencies

It is possible that both hardware and software failures may occur during MAGIC operations on either the Sun or the Macintosh. Depending on the sophistication of the Unix user and whether or not the failure was catastrophic (total workstation lock-up), the user may be able to recover all or part of the data generated during the MAGIC session. However, it is the responsibility of the individual site (and, ultimately, the user) to ensure

that regular and complete system backups are made to tape for the purpose of system recovery and restart.

Assuming that the user is in a MAGIC session and the workstation (or a software package residing on it) fails, the following steps may be attempted prior to system restoration from a backup tape:

- a. Close the malfunctioning window and open a new one.
- b. Go through the system to find and delete the offending window identifier.
- c. Reconnect to the H6000 host within 10 minutes to reaccess open files from the user's Available File Table (AFT).
- d. Reboot the workstation.

5.7 Assumptions and Constraints

A complete discussion of all appropriate assumptions and constraints that are relevant to this effort have been previously discussed in paragraph 2.5.

SECTION 6. SECURITY

This section provides a discussion of security-related considerations as they relate to MAGIC. The following paragraphs provide pertinent background information as well as a discussion of control points, vulnerabilities, safeguards, system monitoring, and auditing.

6.1 Background Information

MAGIC is released as an unclassified system and all system files released with it are unclassified. However, MAGIC's features may be used to analyze and present classified information from classified databases. Under these circumstances, MAGIC must provide the facilities to properly label the screen images and the hardcopy reports, but it is and will remain the user's responsibility to safeguard any and all classified information. MAGIC cannot grant access to classified databases unless the user has permission and access to those files.

Security requirements for all hardware suites and configurations capable of executing MAGIC will remain the same as required for other operational considerations pertinent and applicable to that equipment and environment. Furthermore, the safeguarding of privacy act information also remains the user's responsibility.

6.2 Control Points, Vulnerabilities, and Safeguards

The following subparagraphs will describe MAGIC's control points, the vulnerabilities at the control points, and the safeguard requirements to reduce the risk to acceptable limits.

6.2.1 Control Points. The following subparagraphs will describe MAGIC's input, process, and output control points.

6.2.1.1 Input Control Points. MAGIC's input control points are listed below:

- a. Input data for use within MAGIC may originate from a number of locations: accessible files on the H6000 host, ASCII files in the user's personal directory, ASCII files that are accessible to the user elsewhere on the workstation, Oracle data tables accessible by the user, or ad hoc data entry by the user in a MAGIC user session.
- b. Both a standard keyboard and a three-button mouse are planned for data entry usage in MAGIC
- c. Once entered into MAGIC, source data is kept in standard ASCII-based Unix files of either a temporary or permanent nature depending on whether the user has saved the data.

- d. Whenever data is being input into MAGIC, data validation will be performed to detect errors, report them to the user, and permit interactive correction.

6.2.1.2 Process Control Points. Due to MAGIC's interactive nature, the process control points occur almost immediately after a user requests an operation:

- a. Whenever MAGIC attempts to execute the user's instructions, the system will notify the user if a failure has occurred and why. Success is implicitly understood if no message is provided in accordance with normal Motif-compliant systems.
- b. Whenever a requested operation requires the resources and/or services of a COTS package, MAGIC will activate an interface with that system and support data transfer.

6.2.1.3 Output Control Points. The following output control points are planned for MAGIC:

- a. The workstation screen and any connected hardcopy device is authorized to receive output.
- b. Output products will be distributed and disposed in accordance with all applicable security procedures.

6.2.2 Vulnerabilities. The following vulnerabilities exist for the control points described in subparagraph 6.2.1 above:

- a. As an interactive system, MAGIC is inherently vulnerable to a user entering erroneous and/or invalid data for the context in which the data is to be used. While MAGIC can and will perform data validation operations, context is within the user's scope of responsibility and MAGIC may or may not intercept it.
- b. A sophisticated Unix user can short-circuit MAGIC's normal safeguards and may possibly cause hardware and/or software failure by performing an incompatible operation not designed for.
- c. MAGIC has no method of determining the validity of a user's data access privileges. A valid user name, password, and access rights to data files as enforced by Unix, COTS packages, and/or workstation-resident security software is assumed by MAGIC to be sufficient proof of access privilege.
- d. Beyond verifying a user's desire to delete data files, file linkages, and data descriptions through a dialogue box, MAGIC has no way of stopping a user from erroneously destroying valuable data.

- e. MAGIC assumes no responsibility for user compliance with applicable security requirements regarding generated output. It is the user's responsibility to comply with security regulations.

6.2.3 Safeguards. The following subparagraphs describe MAGIC's administrative, physical, and technical safeguard requirements.

6.2.3.1 Administrative Safeguards. MAGIC will have the following administrative safeguards:

- a. MAGIC system development will be unclassified. However, personnel assigned to develop those portions requiring interaction with the WWMCCS host or any other classified interface will possess an appropriate level of security clearance.
- b. Access permissions to system data and functions will be controlled through established user name/password and file permissions procedures.

6.2.3.2 Physical Safeguards. Physical limitations to data access vary depending on whether or not the workstation involved is linked to the WWMCCS host platform. If connection is possible, standard WWMCCS/WIN physical security procedures dictate that the workstation must be kept in a vaulted, alarmed area.

If the workstation cannot be linked to a classified platform, then standard physical security procedures apply. It must be kept in an office area that may be locked up during non-business hours and checked daily in accordance with normal Close-of-Business (COB) security procedures.

6.2.3.3 Technical Safeguards. The following technical safeguards apply to MAGIC:

- a. User access:
 - (1) All workstations have a System Administrator that generates and issues specific user names/passwords to each user
 - (2) Workstations that are connected to classified platforms or used to process classified data require users to have passwords that are classified.
- b. Process safeguards:
 - (1) MAGIC will perform a number of data validation checks to provide greater data integrity
 - (2) The Oracle RDBMS used by MAGIC uses extensive data encryption to assure data integrity and privacy.

- c. A workstation configured for WWMCCS, WIN, and/or JOPES usage will have automated security identification labeling and display requirements.

6.3 System Monitoring and Auditing

The requirements for the production of an audit trail including automated reports or journals needed to monitor MAGIC are described in the following subparagraphs.

6.3.1 Journalizing. This subparagraph describes all of MAGIC's journalizing requirements in terms of their triggering criteria, identification information, application data, and journal use.

6.3.1.1 Triggering Criteria. MAGIC will use two types of journals that will collect distinctly different information: a system usage journal and a user log journal.

The first will collect its information automatically and is triggered whenever a user launches the MAGIC application. The second journal requires that the user consciously choose to activate the user log. Once activated, this journal will take advantage of the fact that MAGIC is an event-driven system (as are all applications in an X Windows environment). Therefore, user log journal entries will be triggered by the interception of an X event--providing a virtually complete record of a MAGIC user's session.

6.3.1.2 Identification Information. The system usage journal record will, at a minimum, be comprised of five data elements:

- a. Site ID - the installation identification code as available through the WIN system (e.g., NMCC or NMCC2)
- b. Workstation ID - the identification code assigned to the individual workstation by the local System Administrator or security office
- c. Date - the format of the date is immaterial as long as all sites report in the same format
- d. Time - preferably Zulu time but local time is acceptable providing all sites report in local time
- e. User ID - the user's workstation name without the accompanying password.

The user log journal record will, at a minimum, be comprised of two data elements:

- a. Event ID - the code identifying which X event has occurred that triggered the journal record

- b. Corollary Data - any additional event-related data needed to maintain completely accurate journal records of a user's session.

6.3.1.3 Application Data. Only the user log journal record may possibly contain application systems data and that would be limited to a single data element (COROLLARY DATA). The actual data to be recorded is unknown at this time and would be dependent upon the actual X event being recorded.

6.3.1.4 Journal Use. The system usage journal will be used to determine levels and frequency of MAGIC usage among sites where it is installed. As such, some capability to upload the data from individual workstations to the WWMCCS host for transmittal to JNGG will be required. The nature of this capability (manual or automated) is unknown at this time and is dependent on the logistic, difficulty, and reliability factors in providing such a capability. Once available to JNGG, the data will be collected and reduced into a periodic report of system usage by site.

The user log journal is local to the specific workstation. The journal can be used to re-enact the specific MAGIC user session in an automated fashion and the journal will simulate the user's responses exactly. The user may store the journal file to a personal directory for later use or destroy it as any other data file. Since the journal will contain user responses, it is possible that the the journal itself could become classified and normal security procedures for classified data would be required.

6.3.2 Audit Trail. Currently, MAGIC has no requirements for an audit trail although such a requirement may be provided as part of an expanded system usage journal record. If added, at a minimum, a data element for transactions counts would be required.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 7. SYSTEM DEVELOPMENT PLAN

A Software Development Plan (SDP) has been published for MAGIC and is available under separate cover as SDP 2-90. Requests for copies must be forwarded to the Director, DSSO, ATTN: JNGG, The Pentagon, Washington, D.C. 20301-7010.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 8. COST FACTORS

Cost factors in terms of complete life-cycle (development and maintenance) costs as well as risk analysis for the MAGIC efforts is presented and discussed in this section. Since MAGIC is proposed for development in Standard C rather than Ada, both alternatives are analyzed.

8.1 The REVIC Model

Since the cost factors analyses described by DoD Instruction (DoDI) 7041.3 are not particularly applicable to MAGIC's needs, the analysis has been accomplished by using Ray's Enhanced Version of Intermediate COCOMO (REVIC) Cost Estimating Model.

The REVIC Model predicts the development life-cycle costs for software development from requirements analysis through completion of the software acceptance testing and the maintenance life-cycle for fifteen years. It is similar to the intermediate form of the Constructive Cost Model (COCOMO) described by Dr. Barry W. Boehm in his book Software Engineering Economics. The intermediate COCOMO Model provides a set of basic equations that calculate the effort (manpower in man-months and hours) and schedule (elapsed time in calendar months) to perform a typical software development project based on an estimate of the lines of code to be developed and a description of the development environment.

8.1.1 REVIC vs. COCOMO. The primary difference between REVIC and COCOMO is the basic coefficients used in the equations and REVIC's addition of a mode for Ada programs. REVIC has been calibrated to a database of recently completed DoD projects and uses different coefficients. On the average, the values predicted by the basic effort and schedule equations will be higher in REVIC versus COCOMO. The Air Force's Contract Management Division published a study validating the REVIC equations using a database (the database collected by the Rome Air Development Center) different from that used for calibration. In addition, the model has been shown to compare to within 2 percent of expensive commercial models.

Other differences arise in the mechanization of the distribution of effort and schedule to the various phases of the development and the automatic calculation of standard deviations (SD) for risk assessment. COCOMO provides a table for distributing the effort and schedule over the development phases based on the size of the code being developed. REVIC provides a single weighted "average" distribution for effort and schedule, along with the ability to let the user vary the percentages in the system engineering and Development, Test and Evaluation (DT&E) phases.

The REVIC Model has also been enhanced by using Program Evaluation and Review Technique (PERT) statistical methods for determining the lines of code to be developed. Low, high, and most probable estimates for each computer program component (CSCIs in MAGIC) are used to calculate the effective lines of code

and the standard deviation. The effective lines of code and standard deviation are then used in the equations rather than the linear sum of the estimates. In this manner, the estimating uncertainties can be quantified and, to some extent reduced. A sensitivity analysis showing the plus and minus three sigmas for effort and schedule is automatically calculated using the standard deviation.

8.1.2 Calibration and Accuracy. The coefficients used in REVIC's Model have been calibrated to a database of recently completed projects (development phases only) by using the techniques described in Dr. Boehm's book, and provides estimates which are within 5 percent of the projects in the database. The REVIC Model was compared to the commercially available System 3 Model (Dr. Randal Jensen's model as distributed by Computer Economics, Inc.) during a study performed for the Department of Defense. After establishing identical environments for both models, the average efforts predicted by both differed by less than 2 percent, while the average schedules differed by 6 percent. The primary reason for the greater schedule variance is probably due to REVIC's (and COCOMO's) level step staffing profile compared to the Jensen Model's smooth Rayleigh curve staffing.

8.2 Sizing Estimation

A prototype version of MAGIC has been developed on the Sun SPARCstation. Thus, sizing estimates for the C language alternative are the easiest to provide. For this purpose, the number of MAGIC's Source Lines of Code (SLOCs) were counted. The C language sizing estimate appears in table 8-1.

Table 8-1 indicates that MAGIC's total number of SLOCs is currently 29,873 and their apportionment among the system's CSCIs are also shown. The percentage of completion factor is based upon the number of functions currently implemented versus those planned for the target system. Based upon these factors, an estimated number of target SLOCs for the C language alternative is calculated at 52,358 (the last column). The estimated SLOCs for the currently undeveloped CSCIs (Graphic Editor and Slide Show) were calculated by estimating an average of those SLOCs projected for the Business Graphics and Geographic Mapping CSCIs. In summary, MAGIC's C language development is approximately 57 percent complete.

The Ada language sizing estimate is based on these calculations. A well known study was done between 1986 and 1988 by NASA's Goddard Space Flight Center. This study determined that Ada language development required approximately 90 percent more coding (SLOCs) than an equivalent FORTRAN-based development. Since C and FORTRAN are roughly equivalent in their sizing needs, the same conversion factor was used in our estimates. The Ada language sizing estimate appears in table 8-2.

Table 8-2 uses the estimated C language sizing and applies the conversion factor noted previously (90 percent increase). This results in an intermediate sizing estimate of 99,480 Ada SLOCs. It is intermediate because specific technical difficulties arise when attempting to develop MAGIC using

Table 8-1. C Language Sizing Estimation

MAGIC CSCIs	SLOCs	% of System	% Complete	Est SLOCs
Human Interface	20,976	70.22%	55.00%	38,138
Data Management	3,925	13.14%	80.00%	4,906
Business Graphics	1,505	5.04%	95.00%	1,584
Geographic Mapping	1,296	4.34%	80.00%	1,620
Graphic Editor	0	0.00%	0.00%	1,602
Slide Show	0	0.00%	0.00%	1,602
Internal Processing	1,905	6.38%	85.00%	2,241
Programmer Utilities	266	0.89%	40.00%	665
TOTAL	29,873	100.00%	57.05%	52,353

Table 8-2. Ada Language Sizing Estimation

MAGIC CSCIs	Estimated C SLOCs	Conversion Factor	Ada SLOCs	Difficulty Adj Factor	Estimated SLOCs
Human Interface	38,138	190.00%	72,462	1.1	79,708
Data Management	4,906	190.00%	9,321	1.0	9,321
Business Graphics	1,584	190.00%	3,010	1.1	3,311
Geographic Mapping	1,620	190.00%	3,078	1.1	3,386
Graphic Editor	1,602	190.00%	3,044	1.1	3,348
Slide Show	1,602	190.00%	3,044	1.1	3,348
Internal Processing	2,241	190.00%	4,258	5.0	21,290
Programmer Utilities	665	190.00%	1,264	1.0	1,264
TOTAL	52,358	190.00%	99,480	1.3	124,976

- The table uses the estimated C language sizing and applies the conversion factor noted previously (90 percent increase). This results in an intermediate sizing estimate of 99,480 Ada SLOCs. It is intermediate because specific technical difficulties arise when attempting to develop MAGIC using Ada (more code must be developed to handle C to Ada interfaces, memory and stack management, and bindings to C language commercial packages). Thus, an adjustment factor has been added to compensate. As the table clearly shows, the Internal Processing CSCI would be enormously affected and require a much larger number of functions that are significantly more complex. When applied to all CSCIs, an estimated Ada language sizing of 124,976 SLOCs results.

8.3 Ada Language Alternative

The following subparagraphs present and summarize the cost factors pertinent to MAGIC development and maintenance in Ada as well as a risk estimate.

8.3.1 Development Costs. The environmental factors used to produce the REVIC Model's calculations for Ada-based development appear as table 8-3. The REVIC Model's calculations used the ADA Software Development Mode, those environmental factors, and a nominal schedule to arrive at an environmental modifier of 0.649 for this effort. Using the sizing data previously presented in table 8-2, the model also estimated total productivity during development--from Preliminary Design Review (PDR) through Formal Qualification Testing (FQT)--to be 309.5 SLOCs per technical staff month (TSM).

In summary, the data contained in table 8-4 indicates that 594.6 TSMs will be required over 47.7 calendar months. Average staff loading is estimated to be 12.5 Full-Time Support Personnel (FSP) with a peak load of 16.2. The total number of direct labor hours for this effort is estimated to be 92,750 at a cost of \$4,173,754.

8.3.2 Maintenance Costs. For this analysis, an inflation estimate of 5 percent for each year of the life cycle was used. When applied to the estimated base cost of \$45.00 per staff hour and using an annual change traffic estimate of 12 percent, REVIC estimated the total maintenance costs to be \$6,716,292 over 15 years. Staffing requirements fell from 5.0 in Year 1 to 3.4 in Years 4 through 15. Table 8-5 depicts the estimated maintenance costs for Ada.

8.3.3 Risk Analysis. As noted previously in subparagraph 8.1.1, the REVIC Model also provides an automatically calculated sensitivity analysis that uses the standard deviation (also calculated automatically) to show the plus and minus three sigmas for effort and schedule. Table 8-6 uses a standard deviation of 3.471 Thousands of Delivered Source Instructions (KDSI) to conclude that the cost of Ada development may range as low as \$3.95 million to as high as \$4.395 million and take between 45.2 and 50.2 calendar months to complete.

Table 8-3. Environmental Factors for Ada Development

ENVIRONMENTAL FACTOR	RATING	VALUE
Analyst Capability	VH	0.71
Programmer Capability	HI	0.86
Applications Experience	HI	0.91
Virtual Machine Experience	HI	0.90
Prog. Language Experience	NM	1.00
Execution Time Constraint	NM	1.00
Main Storage Constraint	HI	1.06
Virt. Machine Volatility	NM	1.00
Computer Turnaround Time	VL	0.79
Requirements Volatility	NM	1.00
Product Reliability	HI	1.15
Database Size	NM	1.00
Product Complexity	VH	1.30
Required Reuse	VH	1.30
Modern Programming Practices	HI	0.91
Use of S/W Tools	XH	0.73
Required Security	UN	1.00
Mgmt Reserve for Risk	LO	1.20
Required Schedule	NM	1.00
Software Development Mode	ADA	1.00

Table 8-4. Estimated Development Costs Using Ada

	Effort	Schedule	FSP	Costs
Systems Engineering	48.9	9.1	5.4	\$343,048.00
Preliminary Design	93.7	11.8	8.0	\$657,509.00
Critical Design	118.1	7.5	15.6	\$829,033.00
Code & Debug	89.6	4.5	19.8	\$628,922.00
Integrate & Test	105.9	6.3	16.7	\$743,271.00
Dev Test & Integ	138.5	8.5	16.4	\$971,970.00
TOTALS	594.7	47.7		\$4,173,753.00

Table 8-5. Estimated Maintenance Costs Using Ada

YEAR	EFFORT	FSP	INFL	RATE	COST
1	60.6	5.0	5.0%	\$47.25	\$446,583.00
2	52.5	4.4	5.0%	\$49.61	\$406,391.00
3	46.4	3.9	5.0%	\$52.09	\$377,474.00
4	40.4	3.4	5.0%	\$54.70	\$344,650.00
5	40.4	3.4	5.0%	\$57.43	\$361,883.00
6	40.4	3.4	5.0%	\$60.30	\$379,977.00
7	40.4	3.4	5.0%	\$63.32	\$398,976.00
8	40.4	3.4	5.0%	\$66.49	\$418,925.00
9	40.4	3.4	5.0%	\$69.81	\$439,871.00
10	40.4	3.4	5.0%	\$73.30	\$461,865.00
11	40.4	3.4	5.0%	\$76.97	\$484,958.00
12	40.4	3.4	5.0%	\$80.81	\$509,206.00
13	40.4	3.4	5.0%	\$84.85	\$534,666.00
14	40.4	3.4	5.0%	\$89.10	\$561,399.00
15	40.4	3.4	5.0%	\$93.55	\$589,469.00
			TOTAL COST		\$6,716,293.00

Table 8-6. Ada Risk Analysis

	-3 SIGMA	NOMINAL	+3 SIGMA
KDSI	115.6	126.0	136.4
TSMs	562.8	594.6	626.1
Schedule	45.2	47.7	50.2
Total Labor Hrs	87,798	92,750	97,678
Total Costs	\$ 3,950,904	\$ 4,173,754	\$ 4,395,520
STANDARD DEVIATION = 3.471 KDSI			

8.4 C Language Alternative

The following subparagraphs present and summarize the cost factors pertinent to MAGIC development and maintenance in C as well as a risk estimate.

8.4.1 Development Costs. The environmental factors used to produce the REVIC Model's calculations for C-based development appear as table 8-7. The REVIC Model's calculations used the SEMIDETACHED Software Development Mode, those environmental factors, and a nominal schedule to arrive at an environmental modifier of 0.423 for this effort. Using the sizing data previously presented in table 8-1, the model also estimated total productivity during development--from PDR through FQT--to be 380.1 SLOCs per TSM.

In summary, the data contained in table 8-8 indicates that 202.8 TSMs will be required over 34.1 calendar months. Average staff loading is estimated to be 5.9 FSP with a peak load of 7.7. The total number of direct labor hours for this effort is estimated to be 31,631 at a cost of \$1,423,395.

8.4.2 Maintenance Costs. For this analysis, the same inflation factor (5 percent) was used. When applied to the estimated base cost of \$45.00 per staff hour and using an annual change traffic estimate of 10 percent, REVIC estimated the total maintenance costs to be \$1,908,740 over 15 years. Staffing requirements fell from 1.4 in Year 1 to 1.0 in Years 4 through 15. Table 8-9 depicts the estimated maintenance costs for C.

The estimated change traffic was lower for this alternative because of the significantly higher percentage of COTS packages in the C version vice the Ada version.

8.4.3 Risk Analysis. As noted previously in subparagraph 8.1.1, the REVIC Model also provides an automatically calculated sensitivity analysis that uses the standard deviation (also calculated automatically) to show the plus and minus three sigmas for effort and schedule. Table 8-10 is a standard deviation of 1.611 KDSI to conclude that the cost of C development may range as low as \$1.324 million to as high as \$1.523 million and take between 31.7 and 36.5 calendar months to complete.

8.5 Conclusions

The following four conclusions can be made based on the preceding cost factors:

- a. The Ada alternative for MAGIC is projected to be 293 percent more costly than developing MAGIC in C (\$4,173,754 vice \$1,423,395) and require nearly 140 percent more time (47.7 vice 34.1 months).
- b. The Ada alternative for MAGIC is projected to be 352 percent more costly to maintain over 15 years than a C-based MAGIC (\$6,716,292 vice \$1,908,740).

Table 8-7. Environmental Factors for C Development

ENVIRONMENTAL FACTOR	RATING	VALUE
Analyst Capability	VH	0.71
Programmer Capability	HI	0.86
Applications Experience	HI	0.91
Virtual Machine Experience	HI	0.90
Prog. Language Experience	NM	1.00
Execution Time Constraint	NM	1.00
Main Storage Constraint	HI	1.06
Virt. Machine Volatility	LO	0.87
Computer Turnaround Time	VL	0.79
Requirements Volatility	NM	1.00
Product Reliability	HI	1.15
Database Size	NM	1.00
Product Complexity	HI	1.15
Required Reuse	HI	1.10
Modern Programming Practices	HI	0.91
Use of S/W Tools	XH	0.73
Required Security	UN	1.00
Mgmt Reserve for Risk	LO	1.20
Required Schedule	NM	1.00
Software Development Mode	SD	1.00

Table 8-8. Estimated Development Costs Using C

	Effort	Schedule	FSP	Costs
Systems Engineering	16.7	6.5	2.6	\$116,991.00
Preliminary Design	31.9	8.4	3.8	\$224,233.00
Critical Design	40.3	5.4	7.5	\$282,729.00
Code & Debug	30.6	3.2	9.4	\$214,484.00
Integrate & Test	36.1	4.5	8.0	\$253,481.00
Dev Test & Integ	47.2	6.0	7.8	\$331,476.00
TOTALS	202.8	34.0		\$1,423,394.00

Table 8-9. Estimated Maintenance Costs Using C

YEAR	EFFORT	FSP	INFL	RATE	COST
1	17.2	1.4	5.0%	\$47.25	\$126,917.00
2	14.9	1.2	5.0%	\$49.61	\$115,494.00
3	13.2	1.1	5.0%	\$52.09	\$107,277.00
4	11.5	1.0	5.0%	\$54.70	\$ 97,948.00
5	11.5	1.0	5.0%	\$57.43	\$102,846.00
6	11.5	1.0	5.0%	\$60.30	\$107,988.00
7	11.5	1.0	5.0%	\$63.32	\$113,387.00
8	11.5	1.0	5.0%	\$66.49	\$119,057.00
9	11.5	1.0	5.0%	\$69.81	\$125,009.00
10	11.5	1.0	5.0%	\$73.30	\$131,260.00
11	11.5	1.0	5.0%	\$76.97	\$137,823.00
12	11.5	1.0	5.0%	\$80.81	\$144,714.00
13	11.5	1.0	5.0%	\$84.85	\$151,950.00
14	11.5	1.0	5.0%	\$89.10	\$159,547.00
15	11.5	1.0	5.0%	\$93.55	\$167,525.00
			TOTAL COST		\$1,908,742.00

Table 8-10. C Risk Analysis

	-3 SIGMA	NOMINAL	+3 SIGMA
KDSI	48.0	52.8	57.6
TSMs	188.6	202.8	217.1
Schedule	31.7	34.1	36.5
Total Labor Hrs	29,423	31,631	33,864
Total Costs	\$ 1,324,016	\$ 1,423,395	\$ 1,523,873
STANDARD DEVIATION = 1.611 KDSI			

- c. The cost projections for an Ada-based MAGIC are projected to be over 215 percent more risky than the equivalent C-based factors (standard deviations of 3.471 KDSI vice 1.611 KDSI).
- d. MAGIC's C language development is approximately 57 percent finished (noted in paragraph 8.2) with a working prototype available on the Sun SPARCstation.

Therefore, it is deemed more cost-effective to request a waiver from DoD's Ada language requirements for MAGIC and to complete full system development using Standard C.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A
FUNCTIONAL SYSTEM REQUIREMENTS

APPENDIX A. FUNCTIONAL SYSTEM REQUIREMENTS

This appendix provides a list of functional system requirements to be satisfied by the development of MAGIC. These requirements are intended to aid the reader in understanding MAGIC's overall system functionality.

Each requirement has been placed under one of the eight proposed CSCIs. Requirement placement is based on a logical interpretation of the functionality being defined:

A. Human Interface

1. Initialize MAGIC
2. Parse user selections, choices, and language input
3. Recognize interactive operating system commands (i.e., JDAC & TSS commands)
4. Define and name a group of GIPSY statements which may be subsequently executed through reference to the defined name via an application (i.e., DO PROCESS)
5. Execute a GIPSY application which prompts the user for input
6. Perform interactive error detection and handling
7. Provide an on-line interactive help facility
8. Allocate and initialize the DAFC
9. Establish workstation environment to X Windows
10. Access statistics file
11. Use OSF/Motif to provide the graphical user interface.

B. Data Retrieval

1. Identify the user's data file
2. Describe data records
 - a. FDT
 - b. Adding to the Index File
 - c. Augmenting an existing File

- (1) Extended fields
 - (2) Global fields
 - (3) Qualify fields
 - (4) Specific field references.
- 3. Identify any conditional expressions
- 4. Identify any arithmetic expressions
- 5. Retrieve data
- 6. Modify data
 - a. In-line modification
 - b. User subroutine modification
 - c. Record output table
 - d. Field table.
- 7. Manipulate data
 - a. Modify QDT
 - b. Add new fields
 - c. Sort QDF (also resort)
 - d. Qualify data
 - e. Field table QDF
 - f. Field table qualify
 - g. Field table call.
- 8. Populate a database or a data file.
- C. Statistical Graphics
 - 1. Build a tabular report
 - 2. Display a tabular report
 - 3. Modify a tabular report

4. Enhance a tabular report
5. Save a tabular report
6. Access previously saved tabular report
7. Build a new report
 - a. Assign function
 - b. Delete function
 - c. Rename function
 - d. Subset function
 - e. Change function
 - f. Define function
 - g. Add function
 - h. Input function
 - i. Review function.
8. Create and display graphic reports including:
 - a. Bar graphs
 - b. Histograms
 - c. Point graphs
 - d. Line graphs
 - e. Curve graphs
 - f. Step graphs
 - g. Gantt charts
 - h. Pie charts.
9. Modify a graphic report
 - a. Limiting rows, columns, sections, categories
 - b. Adding report totals

- c. Vector sequencing.
 - 10. Display the classification of graphic report
 - a. Report titles
 - b. Clear classification.
 - 11. Explode the wedges in a pie chart
 - 12. Stack the bars in a bar graph
 - 13. Enhance graphic reports with symbols and text
 - 14. Control graph features including size, color, shading, and style of line
 - 15. Save a graphic report
 - 16. Save plotted output.
- D. Geographic Mapping
- 1. Define the map
 - a. Map file
 - b. Map file details
 - c. Map area
 - d. Map projection.
 - 2. Build geographic display
 - a. Grids
 - b. Symbols
 - c. User-defined characters
 - d. Track plot.
 - 3. Generate geographic display
 - 4. View geographic display
 - 5. Modify geographic display
 - 6. Save geographic display.

E. Graphic Editor

1. Compose drawings

a. Standard draw functions

- (1) Freehand drawing
- (2) Create a line
- (3) Create a rectangle
- (4) Create a circle
- (5) Create a polygon
- (6) Create an arc
- (7) Create an ellipse
- (8) Fill.

b. Rubberband technique draw functions

- (1) Create a symbol
- (2) Zoom
- (3) Unzoom
- (4) Pan
- (5) Unpan
- (6) Erase
- (7) Change attributes (e.g., background color, foreground color, line width, line style, fill pattern, fill color).

2. Manage text

- a. Enter horizontal text
- b. Enter vertical text
- c. Enter centered text
- d. Set tab
- e. Clear tab

- f. Rotate text
 - g. Cut text
 - h. Uncut text
 - i. Copy text
 - j. Paste text
 - k. Erase text
 - l. Un-erase text
 - m. Set margins
 - n. Overwrite text
 - o. Change current text attributes (e.g., text size, text style, text spacing, text color, background color).
3. Manipulate an object
- a. Cut an object/symbol
 - b. Uncut an object/symbol
 - c. Copy an object/symbol
 - d. Paste
 - e. Group an object into a symbol
 - f. Split an object from a symbol
 - g. Erase an object/symbol
 - h. Un-erase an object/symbol
 - i. Scale an object/symbol
 - j. Rotate an object/symbol
 - k. Modify object attributes (e.g., object color, line width, line style, fill pattern, fill color).
4. Print functions
- a. Add a slide to an output device queue

- b. List an output device queue
- c. Delete a slide from an output device queue
- d. Reorder an output device queue
- e. Cancel current print job.

F. Slide Show

1. Modifying the slide/briefing inventory

- a. Rename slides
- b. Rename briefings
- c. Create briefings
- d. Delete briefings
- e. Include a slide from the slide inventory to a briefing
- f. Delete a slide from a briefing
- g. Delete a slide from the slide inventory (and from any briefings that may contain it)
- h. Save a slide to the slide inventory.

2. Displaying slides

- a. Display any individual slide from the inventory of slides
- b. Display any individual slide from a briefing
- c. Display the next slide
- d. Display the previous slide
- e. Automatically display all slides in a briefing
- f. Overlay any individual slide over the currently displayed slide.

3. Importing or exporting slides

- a. Copy slide
- b. Copy briefing
- c. Create backup copy

- d. Transfer slide to another inventory
 - e. Transfer briefing to another inventory.
- 4. Printing slides
 - a. Add slide to queue
 - b. List queue
 - c. Delete from queue
 - d. Reorder queue
 - e. Cancel current print job.

G. Internal Processing

- 1. Perform file management
- 2. Control input/output operations
- 3. Save and recall all information necessary to start a new GIPSY session from the departure point of the current GIPSY session (i.e., the DAFC)
- 4. Convert the qualified data and its internal matrix version (i.e., the GDS) to Data Interchange Format (DIF)
- 5. Generate a metafile in a standardized format (e.g., CGM or X bitmap)
- 6. <DELETED>
- 7. Control various devices such as terminals, printers, or plotters via device drivers
- 8. Request operating system services
- 9. Perform specialized processing by executing system-supplied and user subroutines
- 10. Allowing certain globals to prevail throughout a user session
 - a. Command line options
 - b. Module transfer
 - c. Classification markings
 - d. Report titles and modifying them

- e. Clearing specific commands (i.e., CLEAR CLASS)
- f. Size of text
- g. Color of text.
- 11. Color processing
- 12. User control of operating environment attributes
- 13. Identifying GIPSY's collective input statements
 - a. PCS
 - b. Clear PCS
 - c. Save PCS
 - d. RETURN statement.
- 14. Identify and save GIPSY's internal data structures
 - a. FDT
 - b. QDF
 - c. QDT
 - d. DAFC
 - e. GDS.
- 15. Executing user-supplied subroutines
- 16. Executing other TSS or JDAC commands.

H. Programmer Utilities

- 1. Provide statistical reports about MAGIC usage
- 2. Generate printed listings of MAGIC source code
- 3. Sort list of source file information
- 4. Identify differences between two MAGIC source code files
- 5. Update MAGIC area/location names
- 6. Update MAGIC user help and error messages

7. Generate design documentation using an automated tool that reads both MAGIC source code and comments (in-line and header block).

THIS PAGE INTENTIONALLY LEFT BLANK

DISTRIBUTION

DSSO Codes	Copies
JTSA-P (Record and Reference Set)	3
JNCR	1
JNGG	24

Addressees

Defense Technical Information Center (DTIC)	
Cameron Station, Alexandria, VA 22304-6145	2

	30

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved

OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 14 June 1991		3. REPORT TYPE AND DATES COVERED Functional Description	
4. TITLE AND SUBTITLE Functional Description of the Mapping and Graphic Information Capability (MAGIC)				5. FUNDING NUMBERS	
6. AUTHOR(S) David W. Hall					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Defense Systems Support Organization (DSSO) Room BF670, The Pentagon Washington, D.C. 20301-7010				8. PERFORMING ORGANIZATION REPORT NUMBER FD 4-91	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) 107150				10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This Functional Description identifies preliminary design considerations, performance requirements, and user impacts. It also defines the requirements to be satisfied and provides the users with a clear statement of the operational capability to be developed. The FD is the basis for mutual understanding between the user and the developer. It outlines the plan for development and implementation of the Mapping and Graphic Information Capability (MAGIC) and provides a summary of cost factors. The FD is divided into six major sections. These sections cover System Summary (Section 2), Detailed Characteristics (Section 3), Design Considerations (Section 4), Environment (Section 5), Security (Section 6), and Cost Factors (Section 8).					
14. SUBJECT TERMS N/A				15. NUMBER OF PAGES 110	
				16. PRICE CODE N/A	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT None		

THIS PAGE INTENTIONALLY LEFT BLANK